On Role Assignment for Participatory Sensing System

by

Anubhuti Garg

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the MCS degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Mobile crowd sensing is one of the most active areas of research. Participatory sensing is part of it in which participants sense their surroundings and collaborate to accomplish a given task. The participants in reference are smartphones. We focus on location dependent tasks and a problem of role assignment. An existing work on the same defines three types of roles for the participants: broadcasters, normal participants and location information receiver. The broadcasters and normal participants turn on their GPS while location information receivers rely on broadcasters to compute their position. The existing work provides a centralized approach which uses greedy algorithm for role assignment. We propose a sorting based algorithm which minimizes 12-25% of the time for medium and large datasets. We also modify the energy model to minimize power consumption of devices. For this we provide a scheme so that only few devices turn on cellular network to contact server as cellular network consumes considerable energy of smartphones.

In the existing approach if new devices join the region then they cannot participate in the ongoing sensing task until server assigns them role during the next localization phase. In addition to this, if device leaves the region then its neighbouring devices may minimize energy needs by changing their role. However, in the current work the algorithm is required to run over entire set of participants for each insertion and deletion of participant. We provide an alternative method to allocate roles adaptively to new participants and change roles for the existing devices when some devices leave the region on fly. This helps to minimize over 95-99.9% time for role assignment compared to existing state of work.

In addition to this, we have also proposed a distributed approach so that devices are self-capable of assigning role to themselves based on local information. This is first work so far to relieve server from the task of role assignment. Besides proposing a method, we have also taken into account the residual energy of smartphones for assigning the role of broadcaster which has not been considered before. Our algorithm takes 70-85% less time compared to centralized approach but consumes 12-15% more energy as it does not provide optimal set of broadcasters which requires global information. All the work has been validated through extensive experiments using both real and synthetic datasets.

Acknowledgements

My most sincere gratitude to Prof. Amiya Nayak for providing me guidance, support, advice and ideas throughout my thesis work. I wholeheartedly thank him for his precious supervision.

My deepest appreciations to Dr. Nishith Goel and his entire family for their unconditional love and support during my stay here. A special word of thanks to my parents, family and friends for keeping my motivation high and supporting me throughout my thesis.

In the end, I would like to thank all my teachers and mentors who have been instrumental in making me capable of pursuing Masters.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Mobile phones have become an indispensable part of our lives. This has attracted researchers to harness its data sensing capabilities and extract valuable knowledge. Most smartphones are embedded with rich set of sensors such as accelerometer, GPS, gyroscope, microphone, camera and interfaces such as WiFi, Bluetooth and other technologies [2]. This has led to number of exciting applications based on mobile phone sensing.

In this thesis we focus on participatory sensing in which participants actively participate in sensing activity and collaborate to accomplish a given task [1]. Participatory sensing supports various applications ranging from health services to environmental monitoring, most of which are dependent on location information. For example, a person might want to know closest hospital or movie theatre and prices for the tickets. For accurate location information, devices depend on GPS which is a major source of power depletion in cell phones. Therefore, a number of researches focus on providing alternatives to GPS usage.

In [23], authors provide a device to device localization scheme to refrain some devices from using GPS and thereby saving their phone's energy. Energy consumption and accurate real time positioning is very crucial for the sensing task. Localization takes into account mobility and relative positioning of devices. For accuracy some devices need to turn on GPS and others depend on them for calculation of their location.

In [19], authors propose a framework, providing an energy model and greedy method for collaborative localization. Such greedy (brute-force) methods are good only when there is small number of participants. We make use of the same framework and provide improvements for it.

1.1 Motivation

Motivated by the applications of participatory sensing and the need of efficient role assignment for collaborative localization in terms of minimum power consumption and time taken, we focus on these parameters to devise effective algorithms for finding which devices must turn on their GPS so that more time can be utilized in localization, sensing and data processing. This helps to even extend it to medium and large data sets.

Consider an application that maintains gas prices at different locations in a locality. For this participating smartphones are required to upload prices for the gas and their location whenever they are at a gas station. The stations on highways are generally used by tourists. In such scenarios the application must ensure that server is capable of assigning role to new participants and adapt to changes quickly and efficiently. Typically we see that datasets are not updated immediately and roles are not assigned or modified whenever devices join or leave the region. Updates are usually kept and roles are assigned to them during next interval. This refrains devices from participating in ongoing sensing activity. Due to large size of participating devices it is also very undesirable to rerun greedy algorithm again and again. So, there is a need to update and assign roles on-fly in time and energy efficient way. This motivated us to explore it more and propose a way to overcome the problem.

We all know that cellular network drains considerable amount of device power. Moreover, the task of role assignment by the server is a very infeasible technique. We must ensure that server is not burdened too much because it has other important tasks to accomplish such as data collection and processing. This motivated us to work on distributed approach to prevent devices from using cellular data to contact server for receiving their roles. Also, devise a method by which devices assign role to themselves based on local information. However in doing so, we ensure that it is time efficient.

1.2 Contributions

In this thesis, we are going to achieve efficient methods for role assignment to mobile devices in terms of minimum energy and time.

We provide better mathematical model to minimize energy of the system we refer. For this we make changes to the power consumption of devices that play the role of broadcasters and location information receivers. In addition to this, we propose a sorting based algorithm to select broadcasters for the collaborative localization that can effectively minimize time for medium and large datasets. The existing approach is greedy which is effective for data sets of small size. However, these days' people are relying more on smartphones. So, it is beneficial to have a method that can be extended to large datasets as well.

The existing research is based on assigning role to a given set of smartphone devices known to the server. In case new devices enter the region then they collaborate from next sensing task. We provide an adaptive approach to assign role to the participating devices on fly such that server does not have to rerun greedy algorithm for each insertion and deletion. This help new device to participate in the ongoing task and modify the role of existing devices affected due to exit of some participants so, as to minimize energy needs. For this we propose incremental insertion, deletion algorithms and finally an adaptive approach to understand when it is suitable to rerun the greedy or sorting based algorithms.

All the scenarios discussed above are based on centralized approach. The server knows location of all participating devices and based on global information it assigns role to them. However, we provide a distributed approach. For this, we first propose an energy model for distributed scenario. We make necessary changes to the power consumption of devices depending on their role. Then propose a time-efficient distributed algorithm so that devices make decision to allocate role to them. We provide correctness and complexity of this as well. However, being a localized approach it does not provide an optimal set of broadcasters. We also make an attempt to distinguish centralized and distributed approaches.

The performances of the proposed solutions are thoroughly evaluated by simulations using real and synthetic datasets of varied densities.

1.3 Organization

This thesis is organized as follows:

- Chapter 2 provides related work on mobile sensing, participatory sensing, participant selection problem, energy saving and different localization methods available in literature. We provide various applications and frameworks where these methods have been deployed.

- Chapter 3 presents broadcaster set selection problem. We discuss existing framework that has been used for our research. We provide energy model and greedy approach for allocating roles to the devices that already exist in literature. We have made improvements by modifying energy model to minimize power consumption and proposed a sorting based algorithm that works well for medium and large data sets. Simulations on real and synthetic data sets illustrate effectiveness of proposed work.

- Chapter 4 provides an incremental approach to assign role to new devices or modify role of devices when participants leave the region. For this we have discussed possible cases and provided incremental insertion, incremental deletion and adaptive algorithms. Once again we have used synthetic datasets to validate our results.

- Chapter 5 provides distributed algorithm to make devices independent of the server for their role assignment. This Chapter discusses necessary terms, algorithm, correctness and completeness of distributed approach. We have compared it with centralized approach and simulated all results using both real and synthetic datasets of varied densities.

- Chapter 6 makes concluding remarks and presents future work.

Chapter 2

Related Work

Mobile crowd sensing (MCS) is one of the fastest growing research areas due to the increasing dependence on smartphones and its powerful applications. Cell phones provide rich resources such as processing power, memory, sophisticated display, browsing capabilities, messaging, navigation and other powerful applications. One of the major advantages of smartphones is its mobility, portability and small size. These phones are also embedded with multiple sensors and interfaces such as Bluetooth, infrared, WiFi. More research is carried out to make efficient use of sensors to gather data about their environment. However there are many barriers that limit the use of smartphones for data collection and sharing such as privacy concerns, law and order of different countries, phone energy saving and absence of economic incentive to encourage people to participate in sensing activities [1]. Many researchers work towards finding ways to solve such problems.

Figure 2.1 provides an architecture which depicts functioning of MCS applications. All users carrying smartphones collect raw data using sensors and process it through local analytics algorithms which is sent back to the server. The backend processes entire aggregated information.

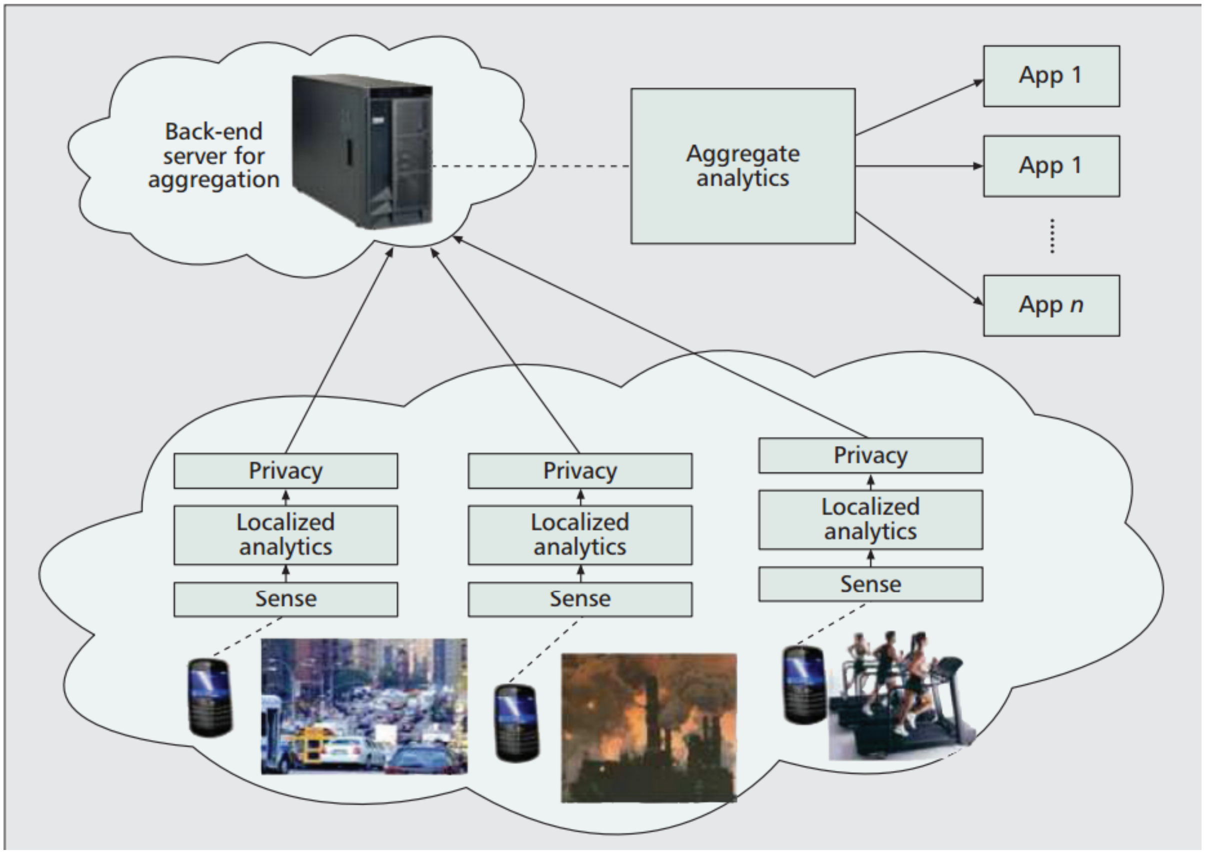

Figure 2.1: Mobile Crowd Sensing Architecture [30]

Technologies that make mobile sensing possible are:

- *Wireless Sensor Networks (WSNs)* which is autonomous sensors spacially distributed to sense environment and cooperatively pass data to the sink. Each sensor consists of radio transceiver with internal or external antenna, microcontroller, battery, connection with other sensors. These are deployed for various applications such as health care, environmental, air pollution, water quality, industrial monitoring, etc.

- *Sensor Webs* are spatially distributed sensor pods that monitor and explore new environments and share information through wireless intra-communication [29]. They are used for understanding spatio-temporal environment. It depends on orbital and terrestrial sensing platforms both of which can be mobile or fixed. Each pod communicates within their local neighbourhood and distributes information. This is in contrast to distributed sensor network which gathers information from one pod and shares it with an uplink point [31].

- *Rich sensor embedded smartphones* are now extensively used for detecting social networks. These smartphones run a user level *Application* configured with *Application Programming Interface (API)* that reads the sensed data using internal sensors and reports them to Web [1].

2.1 Participatory and Opportunistic Sensing

On the basis of node participation, mobile crowd sensing is classified into participatory sensing and opportunistic sensing [2][3]. In opportunistic sensing, decisions are fully automated without any user involvement whereas in participatory sensing, users are actively engaged in sensing activity by determining how, where, what and when to sense. In the following two subsections we discuss applications of both but elaborate more on participatory sensing as it is our main focus in thesis.

2.1.1 Opportunistic Sensing

Opportunistic sensing is free from the burden of selecting users. All the tasks are automated, sensing decisions run in background without letting user know about the active execution of sensing application. These are human-centric because they inherently follow a way so that people opportunistically get into each other's contact [32]. For instance, customer A can build an ad-hoc mobile phone network with another customer B at a shop to collaborate information for helping a tourist.

GeoServ [8] is a protocol where mobile users with smartphones gather location sensitive sensor data and shares it using always-on cellular data connection. Bubble-Sensing [9] is another protocol which opportunistically acquires location based sensing information from the mobile phone users. Recently, Chen et al. [10] proposed distributed algorithm based on stochastic network optimization technique and distributed correlated scheduling for maximizing utility of data collection in smartphones.

2.1.2 Participatory Sensing

Participatory sensing is increasingly gaining interest because of the higher efficiency of data collection. It enables public and professional users to gather, analyze and share local knowledge [33]. It is highly dependent on user's reliability to report data and its compatibility. A great number of participatory sensing applications have been designed and implemented. Figure 2.2 depicts a common architecture of participatory sensing application. Different applications depend on different information to be sensed. The data can be collected in response to some inquiry or for personal use which depends on logged information. Data can be captured in variety of ways. They can be tailored for manual data collection through audio, image capture, context information; or automatic time-location specific sensing through embedded sensors such as GPS, GSM, WiFi, broadband or modifiers such as accelerometer, Bluetooth, etc. The presence of mobile phones with every person irrespective of age, background, society, etc has made it possible to cover entire human population to participate in mobile sensing. The data collected through mobile devices reveal interesting patterns to extract useful information. With advancement of data mining and machine learning, presentation and interpretation of data has enhanced. The server makes use of powerful tools to process aggregated data to serve application requirements.

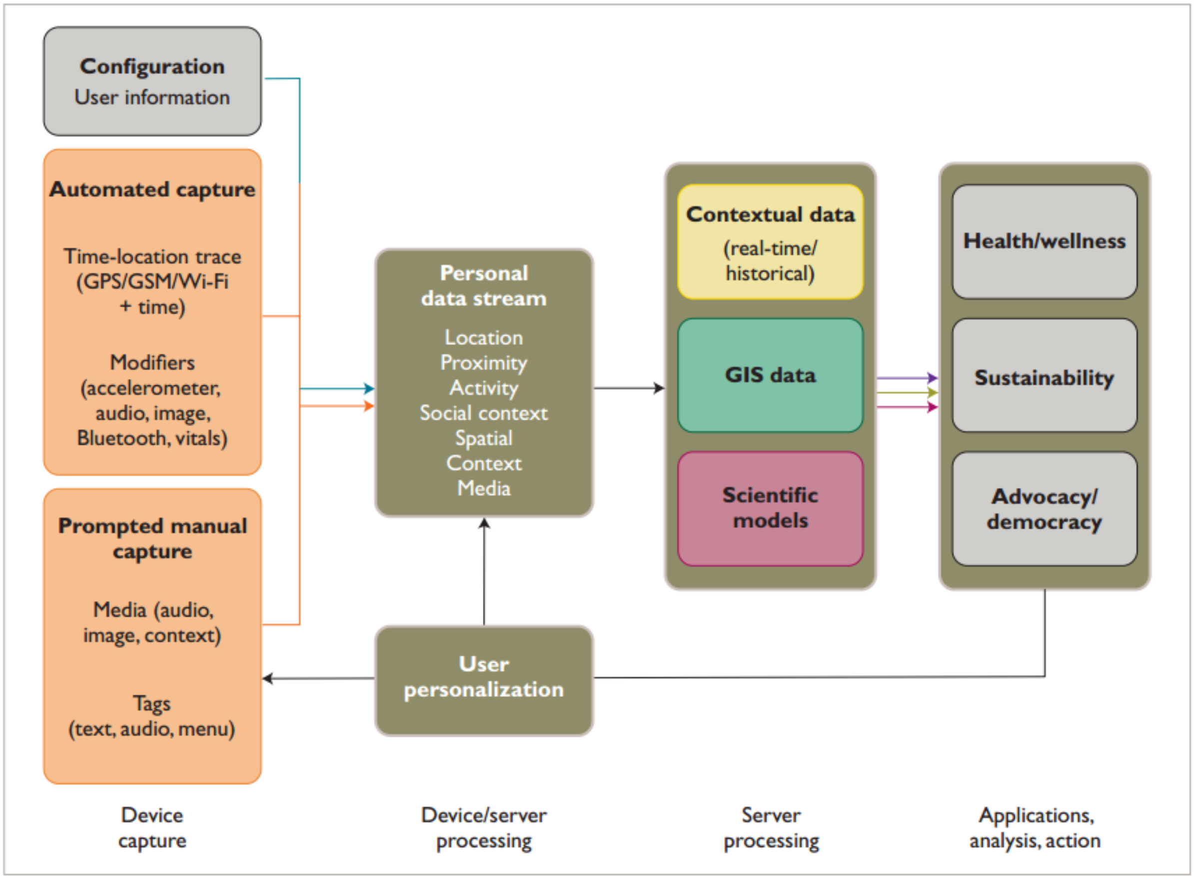

Figure 2.2: Participatory Sensing Architecture [34]

Goldman et al. [35] presented three categories of participatory sensing which we describe below with relevant examples.

- *Collective Design and Investigation* - It is a type of community sensing where a group of individuals come together to decide what, where and why to sense. They collaborate to build a data collection and management system; a method to interpret and process all sensing information; carry out necessary investigation; define privacy and security policies. Here, participants play an active role and are involved at every stage. For instance, consider a scenario where more and more people are being reported to suffer from breathing problems. Then a task can be generated to evaluate air pollution of that region. All residents of this place can participate to sense the pollution level in air. This will help to collect and study the contamination level in different environment such as factory areas, highways, forests, etc. Based on the collected data, essential monitoring steps can be taken. In this scenario, people all over the country are not required to participate. Only residents of a region pool solve the problem together.

- *Public Contribution* - In this category, an external organization or individual defines the sensing problem; data collection and management schemes; and data mining algorithms required to process information. The task is then passed to a set of users to perform the sensing operation. The users involved in sensing are completely free from defining the problem. Researchers are mostly dependent on such type of sensing activity to solve a research problem. They gather data from group of users that might be university students or professionals from different companies, etc. Then process the collected data to extract meaningful information.

- *Personal Use and Reflection* - In this case, individuals log their activity in form of images, videos or applications supported by data management tool. This helps them to seek some pattern and monitor their activity which can be used to either change some habits, analyze health or social dynamics. This is a type of self-discovery tool. There are various applications available on phone to log health data such as sleep and rising time, body measurement, walking and running distance, etc.

Figure 2.3 shows the flow chart of participatory sensing system in action. The participatory sensing opens exciting opportunities and below we discuss few of the important areas where it has been successfully deployed.

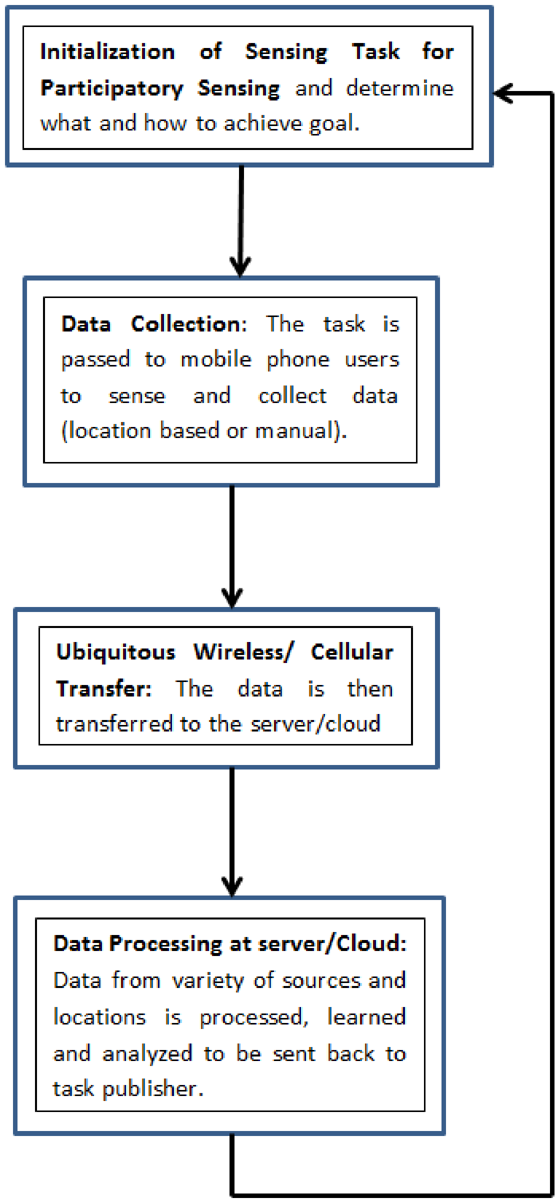

Figure 2.3: Participatory Sensing System in Action

2.1.2.1 Healthcare

Worldwide mobile phones have been successful in empowering medicinal service-experts. They now have admittance to far reaching ongoing patient information for the purpose of health care. All the more critically, clients can ceaselessly and much of the time track their well-being on the go and adjust their ways of life. As of late, there has been a developing enthusiasm for creating proactive well-being related cell phone applications [36]. Indeed mobile phone sensing opens up tremendous potential for healthcare services.

AndWellness [38] was developed as a personal behavioral and contextual data collection system to record and monitor participants daily habits. The system can be deployed to assess HIV+ patients through behaviors and emotional distress. Similarly, Ryder et al. [39] developed a system to evaluate mobility affecting chronic diseases such as MS, Parkinson's, and Muscular Dystrophy. The application starts collecting mobility based data using GPS as soon as phone is turned on. The GPS is switched on only when user is outdoor which is detected using accelerometer.

BikeNet [40] is a sophisticated system developed to evaluate cyclist performance and fitness. It collects time and location based data regarding current speed, average speed, distance travelled, calories burnt and path inclination. It is also capable of recording heart rate and galvanic skin response to extend it to measure emotional excitement and stress level. The system is capable of providing cyclists the healthiest route to take based on the noise, pollution level and terrain roughness. StressSense [41] is another similar system developed to recognize stress from human voice.

2.1.2.2 Environment Monitoring

Common Sense [5] was developed as a mobile sensing technology to gather, analyze and share environmental data from mobile phones. Another similar application is NoiseSpy [6] which uses users' mobile phone to log data for monitoring environmental noise using microphone. This is combined with GPS data to generate map of sound levels. Zhang et al. [7] proposed a cosine theorem based algorithm to fuse and find conflicting data collected by participants.

Biketastic [42] is a system that collects route experience of bikers such as terrain, noise level, scenery image, etc. The data is collected periodically after every one second using GPS. The noise level and roughness is sensed using accelerometer. The information of different routes is made available to cyclists' inorder to facilitate them in choosing right path. Kaenel et al. [43] developed a participatory sensing application to analyze thermal effects in atmosphere using data collected through paraglider pilots. They make use of incentive mechanism and efficiently deal with possible errors that can be encountered in representing data concisely. To cope the unbalanced distribution of measurement in time and space they proposed probability based thermal maps.

2.1.2.3 Urban Sensing

Miluzzo et al. [44] developed a people-centric application that deploys sensors available in mobile phones to infer people's sensing presence and shares it with others through social network portals such as Facebook, etc. The application uses split-level classification to divide the task into two phases - classification at phone level and server level. The output of classification at individual phone is called primitive which is sent to backend sever to extract facts. They have also used the technology of power aware duty-cycling and sleep scheduling strategy to increase battery lifetime.

Numerous customers are misled to paying high prices at stores. Therefore, Deng and Cox [45] proposed a tool for helping customers to save money. They make use of mobile phone cameras to extract meaningful information from two-dimensional barcode which is uploaded to server along with location information to pinpoint the place of the store. Inherently, they rely on incentive scheme to lure customers for participating and suggest methods for data integrity.

Participatory Urbanism project [4] involves participation of urban people to collect and share air quality measured with sensors enabled in mobile device. Peebles et al. [46] presented a framework to model human behavior by exploiting existing classifiers to efficiently extract information even when data submitted by users is unreliably labelled.

2.1.2.4 Transportation and Communication

ParkNet [47] is a framework used to collect parking space occupancy information. Vehicles were considered to be equipped with GPS to detect location and used ultrasonic range finder to determine parking spot occupancy. They have deployed environmental fingerprinting approach to derive real time map of parking availability at the server. This was made available to clients in search of parking space. Similarly, Jakob et al. [48] developed an efficient framework for monitoring road surface. They make use of vibrations and GPS sensors to collect data and simple machine learning approaches to extract useful information. Then they apply clustering to reduce spurious detections and misidentify good road segments.

2.2 Participant Selection Problem in Mobile Sensing

Most of the research under mobile sensing is based on the criteria of selecting appropriate participants for data collection. For instance, Song et al. [11] proposed a constrained optimization problem for selecting participants which can meet quality-of-information (QoI) requirements of the sensing tasks. They have also developed an energy consumption index to measure the impact of performing the task on different smartphones with different energy levels.

Similarly, Tuncay [12] proposed people centric, distributed and autonomous framework for efficient sensing and collection of crowdsourcing data. Only the nodes that visit the sensing location were recruited for collecting the data. For this they compute similarity between single location and user's behavioral profile. Reddy et al. [13] developed a method to recruit participants on the basis of geographic, temporal availability and participation habit. They seek to recruit participants that maximize the coverage and meet certain requirements which are measures of sampling likelihood, quality, etc. But it depends on past coverage and participation behavior. Recently, Li [14] proposed an efficient method for participation recruitment incorporating heterogeneous tasks by minimizing sensing cost and maintaining certain level of probabilistic coverage. They proposed both online and offline greedy algorithms and validated their results with real dataset.

2.3 Energy Saving in Mobile Sensing

Energy saving is quite crucial for mobile sensing as mobile devices have limited battery life and sensing tasks consumes considerable energy. In [15], authors use adaptive pipeline based processing on phones for continuous sensing and monitoring of human activities and context. It is energy efficient and takes into account mobility and behavioral patterns. To validate their work, they deployed it into two applications - JigMe to record person's daily diary and GreenSaw to monitor calorie expenditure and carbon footprint.

PSeense [17] is another application proposed to reduce energy consumption of mobile devices. They make use of the adaptive positioning mechanism and short range communication to exchange position related information. The server sends the set of locations that are needed to be sensed. Each mobile device then periodically fix its location to cover all queried regions. Liu [18] presented an energy efficient participant selection algorithm based on constrained optimization problem and Quality of Information (QoI) satisfaction ratio which is evaluated on the basis of data granularity and quantity. They propose a behavioral model to find relationship between residual energy and willingness to participate so as to know beforehand which participant will deny participating in the sensing activity.

The above discussed work are either dependent on sensing human activities or assume nodes move to collect information from different regions of interest. However, we are concerned with the problem of location-dependent sensing. The tasks that we are interested in need participation of all devices and does not assume any mobility pattern. The same has been proposed by authors in [19]. They presented a collaborative localization technique to minimize energy consumption of smartphones for sensing tasks. Its localization technique is discussed in next section.

2.4 Localization Methods

Traditionally GPS has been used for localizing devices but many alternative methods have been explored and exploited. Authors in [20] provide localization scheme based upon Bluetooth technology to minimize power and provide accuracy. They do not make use of lookup table and service connections. The application performs Bluetooth name resolution and then computes distance to every other device using received signal strength. However, they do not explore energy optimization during localization phase. Xiao [21] provide a method for devices to estimate their position based on the location of single moving beacon using range-free technique. The mobile nodes were considered to be equipped with GPS and other nodes depend on their beacon messages to estimate their location in a fully distributed manner. The beacon messages were periodically sent. Authors make use of Gauss-Markov mobility model to eliminate the need of using stationary anchors.

Zhang [22] provide method to select mobile anchor nodes equipped with GPS. These nodes collect location information which is broadcast periodically for other nodes to estimate their location based on RSSI technique. This enhances energy saving performance but consumes more time for deployment and maintainenance. In [22] authors present a work on Wireless Sensor Networks (WSNs). A sensor node estimates its location using RSSI of beacon messages sent by group of mobile anchor nodes' (GMAN) periodically. The GMAN includes three anchor nodes which are mobile yet their relative locations remain constant. These are positioned in such a way that they form an equilateral triangle. Only one anchor node among the three in GMAN turns on GPS to estimate position. Others calculate their location with respect to it. The GMAN moves at a constant velocity through the deployed area.

In the above work, authors have chosen WSNs to present localization scheme. There has been a shift from WSN to smartphones as they are equipped with more sophisticated technologies and higher battery life. In [16] authors use a strategy to perform localization only when users have finished moving. The framework makes use of periodic polling of accelerometer to detect whether user is walking or not and localize using WLAN technology. However, GPS provides better accuracy of location but drains considerable power. Also, there is a need to incorporate mobility when localizing devices.

Recently, authors in [23] proposed a device-to-device localization scheme. The change of the distance between mobile devices is calculated by the change of the wireless signal strength between them. And use step-counting method to detect movements of devices. Consider the Figure 2.4 shown below. Let d^i, d^j denote the distance between B and C at time t_i and t_j and the movement of C during this interval is shown by $\vec{m}_i$. From the free space radio propagation model [24], the ratio between d^i, d^j can be calculated by:

$$\frac{d^i}{d^j} = 10^{\frac{r^j - r^i}{10n}}$$

where r^i is the RSSI that B receives at time t_i.

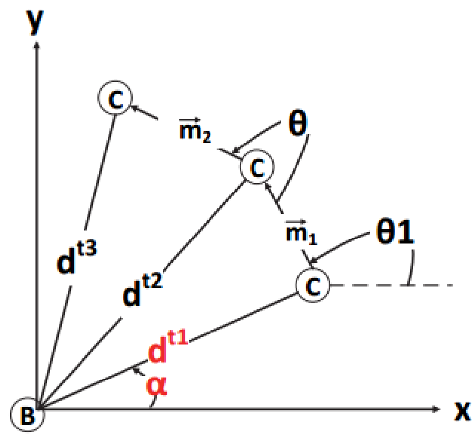

Figure 2.4: Device-to-Device Localization [19]

From this, the distance between B and C at time t_1 can be obtained using:

$$arccos\frac{({k_{t_2}}^2-1)(d^{t_1})^2+(m_1)^2}{2k_{t_2}d^{t_1}m_1}+\theta = arccos\frac{({k_{t_2}}^2-{k_{t_3}}^2)(d^{t_1})^2+{m_2}^2}{2k_{t_2}d^{t_1}m_2}+2\pi$$

where $k_{t_i} = 10^{\frac{r^1-r^i}{10n}}$.

From this the relative angle, α between B and C can be obtained using following:

$$\alpha = arccos\frac{(d^{t_1})^2+{m_1}^2-{k_{t_2}}^2*(d^{t_1})^2}{2(d^{t_1})m_1}$$

This method is used in [25][19]. We make similar use in our work as well.

Chapter 3

Broadcaster Set Selection Problem

In this Chapter we shall present an existing work on participatory sensing system [19] in which every participant carrying smartphone senses its environment and shares it with server. Tasks are such that they need location of devices to sense data. However, if every device makes use of GPS for localization then it is quite inefficient as GPS drains considerable amount of energy. So, a set of devices are chosen as broadcasters which turn on GPS and the neighbouring devices rely on them to calculate their position.

We propose an efficient energy model to minimize the power consumption of such a system. The existing scheme for finding optimal set of broadcasters is based on greedy algorithm. This is time efficient only when the number of participants is small. We propose a sorting based algorithm. This provides better time complexity for moderate and large data sets which is the actual case in real scenarios. We validate our work with extensive experiments on both real and synthetic datasets. Simulation results demonstrate that our proposed approach effectively detects optimal set of broadcasters so, consumes same energy as obtained by greedy algorithm and minimizes time taken for role assignment.

3.1 System Overview

Authors in [19][25] present a framework for collaborative sensing comprising of central server, task publisher and a set of smartphones in a region. The main idea is to find optimal set of broadcasters so as to minimize the energy consumption. The participatory sensing system is shown in Figure 3.1. The task publisher sends sensing task to server

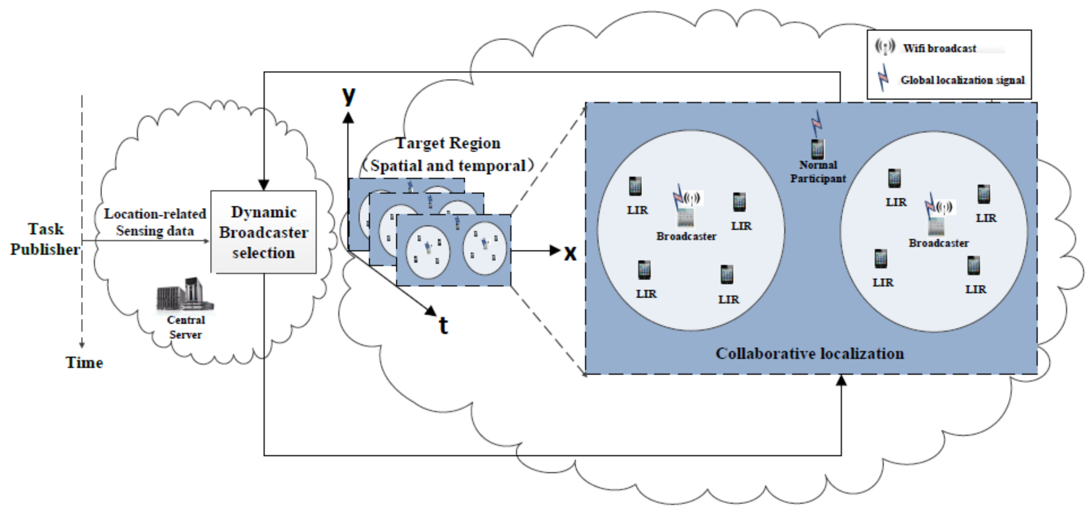

Figure 3.1: System Architecture [19]

which forwards the same to participating smartphones. These participants are assigned roles by server for some time period. There are three category of roles that can be assigned -

- *broadcaster*: its function is to obtain location using GPS and broadcast its location and movement information to the surrounding participants;
- *location information receivers (LIR)*: they rely on broadcasters to calculate their location using device-to-device localization method; and
- *normal participants (NP)*: they do not receive any location broadcast from the surrounding and obtains location through GPS.

The roles last for certain period of time then server assigns new roles according to the updated location information.

The entire duration from task arrival to task delivery is divided into two phases. In the initialization phase, server receives task and notifies all participants. Each one of them then turns on its GPS to obtain initial location. This is shared with server through cellular network which help server to assign them roles. The assigned roles last for a certain period after which roles are reassigned. During the next collaborative localization phase, broadcasters and normal participants turn on their GPS to obtain location for sensing task. Broadcasters broadcast their location and movement information periodically to their surrounding LIRs which calculate their location using device-to-device localization method. All participants regardless of whether they are broadcaster, LIR or NP contact server using cellular network to share their location and sensing data.

3.1.1 Notations

Table 3.1 provides a list of all notations used in this Chapter with their meaning.

3.2 Problem Statement

The goal of our work is to improve the existing energy model [19] so as to minimize the power consumption of system and individual devices. We also aim to improve the method for finding optimal set of broadcasters for a fixed interval (say $[t_1, t_2]$). A method for finding the length of localization period is already provided by authors in [19]. We will state the problem statement then provide the existing energy model.

Table 3.1: List of notations

Notation	Explanation
M	Set of smartphones
$B_{t_1t_2}$	Set of broadcasters during $[t_1, t_2]$
$N_{t_1t_2}$	Set of normal participant for the interval $[t_1, t_2]$
b_m	Boolean to indicate if smartphone m is selected as broadcaster
n_m	Boolean to indicate if smartphone m is selected as normal participant
br	Boolean to indicate if smartphone m is selected as LIR
$P_{t_1t_2}$	Physical connectivity matrix for interval $[t_1, t_2]$
p_{ij}	The physical connectivity between participants i and j
D_t	Distance matrix at time t
e_b	Energy of broadcaster
e_n	Energy of normal participant
e_l	Energy of LIR
e_g	Power of GPS
e_c	Power of cellular network for localization
e_{w1}	Power of WiFi communication for sending
e_{w2}	Power of WiFi communication for receiving
$E_{t_1t_2}$	Energy consumed during $[t_1, t_2]$
E_{ms}	Temporary energy consumed
κ_{max}	Upper bound used in proposed algorithm
κ_i	Connectivity of a node i
κ'_i	Local Connectivity of a node i
G_κ	Global Connectivity of broadcaster set

The broadcaster set selection problem is formulated as:

Minimize: Energy of all participants during $[t_1, t_2]$, $E_{t_1t_2}$

Subject to:

$$b_m = 0, 1, \forall m \in M \tag{3.1}$$

$$p_{ij} = \begin{cases} 0, d_{ij} > a_r - v \times (t_2 - t_1), \forall i, j \in M \\ 1, d_{ij} \leq a_r - v \times (t_2 - t_1), \forall i, j \in M \end{cases} \tag{3.2}$$

$$n_m = \begin{cases} 0, (\sum_{j=1}^{|M|} p_{mj} \geq h, \forall m \in M \setminus B_{t_1t_2}) \vee (\forall m \in B_{t_1t_2}) \\ 1, \sum_{j=1}^{|M|} p_{mj} < h, \forall m \in M \setminus B_{t_1t_2} \end{cases} \tag{3.3}$$

where M denotes a set of smartphones and $B_{t_1t_2}$ denotes set of broadcasters for the interval $[t_1, t_2]$. The Boolean b_m and n_m are used to indicate that smartphone m is selected as broadcaster or normal participant respectively. Eqn. 3.1 is integer constraint. p_{ij} in Eqn. 3.2 represents an element of physical connectivity matrix, $P_{t_1t_2}$ to depict connectivity between participants i and j for the interval $[t_1, t_2]$. It can be determined by current distance matrix d_{ij}, length of period $(t_2 - t_1)$, location accuracy, a_r and moving speed of pedestrian, v. Eqn. 3.3 enforces LIR to receive broadcast from at least h broadcasters to calculate its location. However, we consider $h = 1$ for experiments which can be extended further. Every participant is assumed to have enough power to be chosen as broadcaster.

3.3 Existing Energy Model

In the existing model, e_b is considered as the power of broadcaster for collaborative localization in $[t_1, t_2]$ given by:

$$e_b = ((e_g + e_{w1}) \times (t_2 - t_1) + e_c \times t_c)/(t_2 - t_1) \tag{3.4}$$

where e_g, e_c is the power of GPS and cellular network for localization respectively. e_{w1} denotes the energy consumption of WiFi communication for sending/broadcasting location information to the surrounding. We use t_c to depict the time that participants need for sending data and receiving roles.

The power of LIR is denoted by e_l for collaborative localization in $[t_1, t_2]$, given by

$$e_l = (e_{w2} \times (t_2 - t_1) + e_c \times t_c)/(t_2 - t_1) \tag{3.5}$$

where e_{w2} is used to indicate the energy consumption of WiFi communication for receiving the location information from its neighbouring broadcasters.

Similarly, e_n denotes power of normal participant during $[t_1, t_2]$ given by:

$$e_n = (e_g \times (t_2 - t_1) + e_c \times t_c)/(t_2 - t_1) \tag{3.6}$$

It is worth noting that $e_g > e_c > e_{w1} > e_{w2}$.

Whenever a task arrives, server assigns role for a period $[t_1, t_2]$ where $\forall t_1, t_2 \in T$, T denotes set of time $T = t_s, t_s + 1, t_s + 2, \ldots, t_e$; t_s, t_e are the start and end time of the task. During $[t_1, t_2]$, broadcasters and NP use GPS to obtain location. Then broadcasters use WiFi to broadcast its location and movement information to the surrounding LIRs which apply device-to-device localization to obtain their position. All mobile devices then perform sensing activity. From $[t_2 - t_c, t_2]$ all participants upload their location and sensing data to the server using cellular network. This energy consumption model is shown in Figure 3.2.

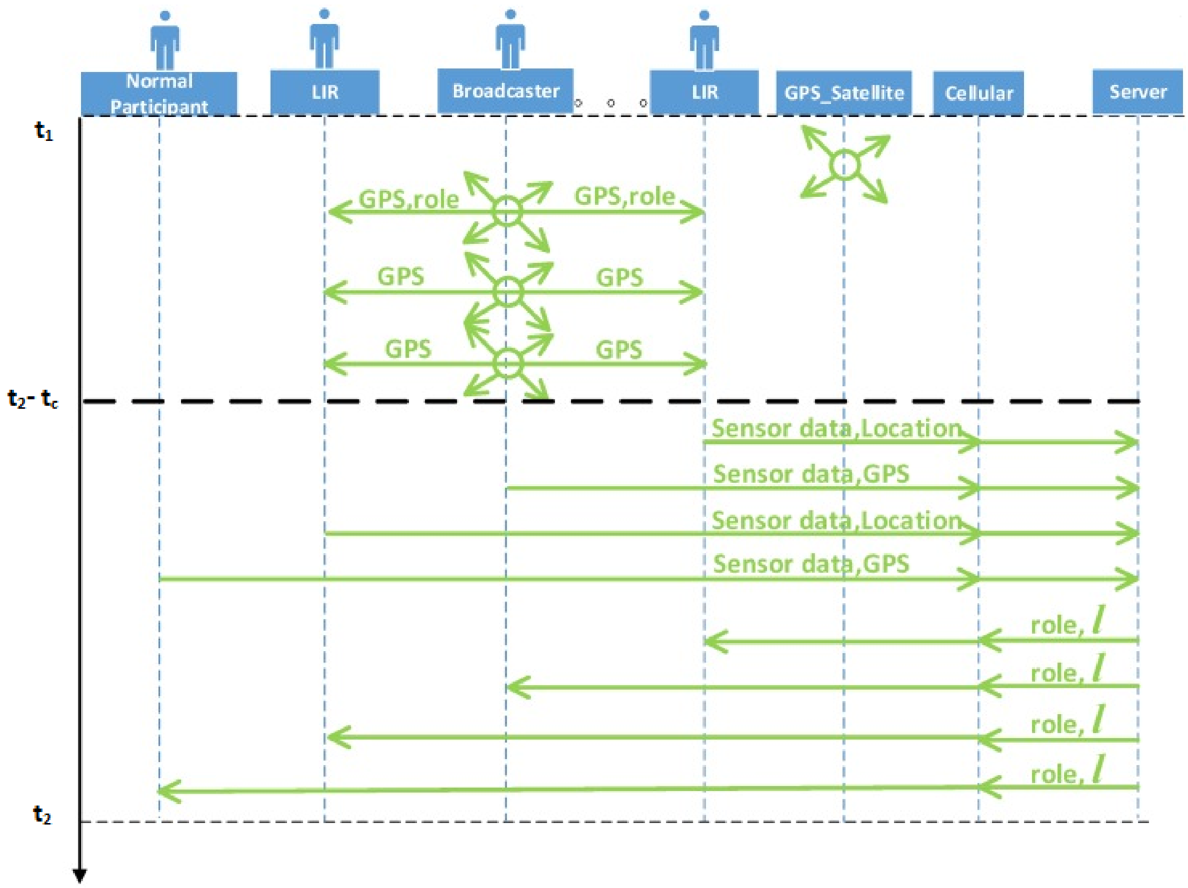

Figure 3.2: The collaborative localization during $[t_1, t_2]$ [19]

Let Boolean b and n indicate mobile participant selected as broadcaster and normal participant respectively during an interval $[t_1, t_2]$. The power of all participants for the interval $[t_1, t_2]$ is given by $E_{t_1 t_2}$ which comprises power of all broadcasters E_b, the power of all LIRs E_l, and power of all normal participants E_n. These are calculated as below.

Given set of all broadcasters $|B_{t_1 t_2}|$, the power consumed by this set of broadcasters for the period can be written as:

$$E_b = |B_{t_1t_2}|e_b = \sum_{m=1}^{|M|} b_m e_b \tag{3.7}$$

Similarly, for the set of normal participants $N_{t_1t_2}$, the power of all normal participants can be calculated by following

$$E_n = |N_{t_1t_2}|e_n = \sum_{m=1}^{|M|} n_m e_n \tag{3.8}$$

Remaining smartphones plays the role of LIRs so, their power can be calculated by:

$$E_l = (|M| - \sum_{m=1}^{|M|} b_m - \sum_{m=1}^{|M|} n_m) \times e_l \tag{3.9}$$

Therefore, the total energy consumption for all participants during the collaborative localization time period $E_{t_1t_2}$ comes out to be:

$$E_{t_1t_2} = E_b + E_l + E_n = \sum_{m=1}^{|M|} b_m e_b + (|M| - \sum_{m=1}^{|M|} b_m - \sum_{m=1}^{|M|} n_m) \times e_l + \sum_{m=1}^{|M|} n_m e_n \tag{3.10}$$

3.4 Improvement of Energy Model

In the existing work, we find that every participant after sensing its environment contacts server using cellular network to share its data and location information. This consumes considerable amount of energy. We extend the role of broadcasters to aggregating location information and data of its LIR devices. This helps to save battery of LIRs by restricting them to turn on their cellular network and contacting server directly.

The aggregation of data at broadcasters is based on the observation that broadcaster set is smallest and has capability of covering LIRs within its range. It adds energy to the broadcasters but since LIR forms the largest set amongst broadcaster and normal participant set so, it helps in saving energy of the system.

The new roles for the next localization are received by broadcasters and normal participants. Broadcasters share it with its previous set of LIRs.

Lemma 1 *Role of LIRs cannot change to normal participants in the next collaborative localization period.*

Proof: Consider that one of the LIRs is assigned a role of NP. This means there does not exist any node within its range. If this would have been the case then LIR could not calculate its location using position of broadcaster. Hence, there exists a contradiction. □

The role of broadcaster can change to normal participant; normal participant can become broadcaster or LIR; and LIR can become broadcaster. In any of the role change, every device will be able to receive new roles from the previous set of broadcasters and normal participants. For this we add another parameter $t_{c'}$ to represent the time needed for LIR to send their information and broadcaster to receive from them. The power consumed if a device is assigned the role of broadcaster, LIR or normal participant is provided below.

Let e_b be the power of a broadcaster during interval $[t_1, t_2]$. It is illustrated in Figure 3.3 and mathematically formulated as:

$$e_b = ((e_g + e_{w1}) \times (t_2 - t_1) + e_{w2} \times t_{c'} + e_c \times t_c)/(t_2 - t_1)$$

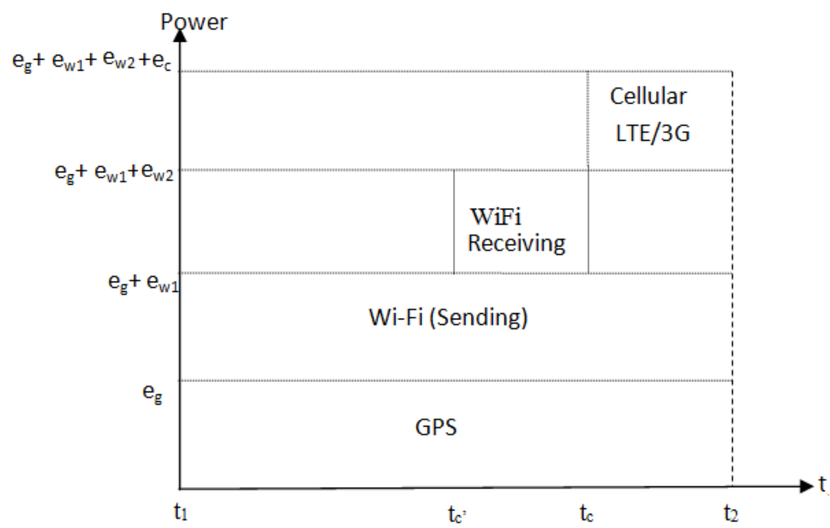

Figure 3.3: Energy Consumption of Broadcasters during $[t_1, t_2]$

Let e_l be the power of LIR for the collaborative localization during $[t_1, t_2]$ which is depicted in Figure 3.4 and mathematically written as:

$$e_l = (e_{w2} \times (t_2 - t_1) + e_{w1} \times t_{c'})/(t_2 - t_1)$$

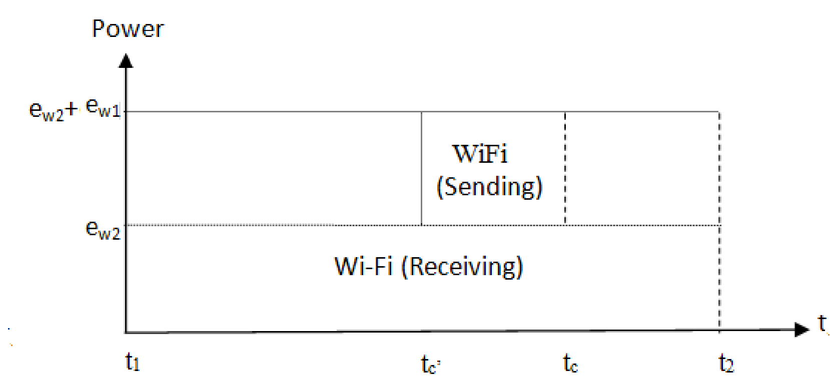

Figure 3.4: Energy Consumption of LIRs during $[t_1, t_2]$.

Let e_n be power of NP during $[t_1, t_2]$. Figure 3.5 shows the energy consumption model for normal participant. We use same equation as presented in [19].

$$e_n = (e_g \times (t_2 - t_1) + e_c \times t_c)/(t_2 - t_1)$$

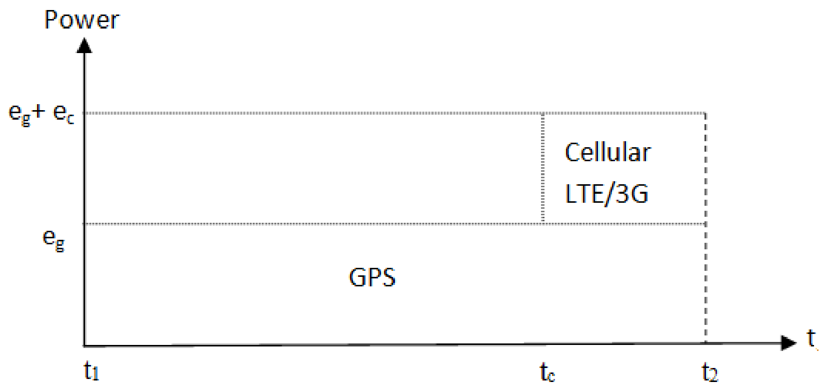

Figure 3.5: Energy Consumption of normal participants during $[t_1, t_2]$.

3.5 Sorting Based Broadcaster-set selection

The authors in [19] provide a greedy algorithm for finding an optimal set of broadcasters. The idea was to check every participant for the role of broadcaster in each iteration and select the one which minimizes the most system energy. The aim of the algorithm is to get optimal set of broadcasters and then filter out normal and LIR participants from the remaining set of devices. We assume that nodes that are physically connected to it are the ones that are within its broadcasting range.

We provide a sorting based solution for selecting broadcaster set. To explain the proposed algorithm, we introduce few definitions.

Definition 1 Connectivity (κ) of a node *is defined as the number of participants within node's boadcasting range. Mathematically, connectivity of the node i can be given as*

$$\kappa_i = \sum_{j=1}^{M} p_{ij} \tag{3.11}$$

Definition 2 Global Connectivity (G_κ) of broadcaster set *is defined as number of participants that can be part of LIR set as a result of the chosen set of broadcasters.*

Definition 3 Local Connectivity (κ') of a node *is defined as the number of devices that can possibly become LIR (apart from the ones that are already part of LIR set) as a result of this node being chosen for the role of broadcaster.*

Lemma 2 *Local connectivity of a node is less than equal to its connectivity.*

Proof: The connectivity parameter includes all devices reachable within the broadcasting range of the node. However, devices which are already covered by selected broadcasters are part of LIR set hence do not contribute to the local connectivity. $\square$

We maintain a Boolean vector, *br* initialized to zero. The vector is updated when a node, i is elected as broadcaster by performing bitwise-OR operation of *br* and p_i. It is assumed that $br(j) = 0, \forall j \in B_{t_1 t_2}$. The global connectivity can be written as:

$$G_\kappa = \Omega(br) \tag{3.12}$$

We introduce an operator $\Omega(\cdot)$ to compute G_κ where $\Omega(t)$ counts the number of ones in given Boolean vector, t. In the above case, number of ones in vector *br* provides number of devices that become LIR as a result of devices chosen for the role of broadcaster.

Similarly, we have used $\oplus$ to represent Boolean-XOR operation and $\vee$ for Boolean-OR operation. The local connectivity of a node i (κ'_i) can be formulated as,

$$\kappa'_i = \Omega((br \vee p_i) \oplus br) \tag{3.13}$$

where $p_i(j) = 0, \forall j \in B_{t_1 t_2}$.

This helps to detect unique set of devices that node i contributes for the role of LIR. It is worth noting that devices are not assigned role LIR until broadcasters are chosen. Even if a node becomes eligible for LIR role, it is still checked for the role of broadcaster. The vector br is only used for calculating global and local connectivity.

The idea is to first sort all the participants on the basis of their connectivity. Then choose topmost candidate in the list as next broadcaster in every round. However, in doing so it does not provide required optimal set of broadcasters. This happens because candidates change their local connectivity. Only the participant having next highest local connectivity becomes part of broadcaster set. One solution for achieving this is to sort participants in every iteration (that is, brute-force search) for selecting next broadcaster. This makes time complexity even worse than greedy solution. To tackle this problem, we sort participants in decreasing order of connectivity; that is, on the basis of number of nodes that are physically connected to each participant. Then, we set an upper bound for every iteration.

Definition 4 *Upper bound (κ_{max}) is a threshold used to limit the search space. It is initialized to κ'_1 (local connectivity of first node) which limits search upto the node having $\kappa > \kappa_{max}$. At any instant, if κ'_j is greater than κ_{max} then κ_{max} is reassigned to κ'_j.*

We assume that once a node is selected as broadcaster, it is removed from the list. Always the first node in the list happens to be the node with highest connectivity. The κ_{max} is set to local connectivity of first node that is κ'_1 where ($\kappa'_1 \leq \kappa_1$). The node having highest κ' is selected as next broadcaster with search space limited upto the node having $\kappa > \kappa_{max}$.

Lemma 3 *The next broadcaster to be chosen exists within the search space limited by upper bound.*

Proof: The search space is limited upto the node having connectivity greater than the bound. It is assumed that nodes are sorted on the basis of their connectivity and setting $\kappa_{max} = \kappa'_1$ initially.
Case 1: All nodes with $\kappa_j < \kappa_{max}$ cannot be become broadcaster as there already exists a node with local connectivity higher than its connectivity (list is sorted). $\kappa'_j \leq \kappa_j \leq \kappa_{max}$.
Case 2: Updating κ_{max} when $\kappa'_j > \kappa_{max}\ \forall j,\ \kappa_j > \kappa_{max}$ further decreases search space and so we need to search only upto nodes having $\kappa_k > \kappa_{max}$. All nodes further down the sorted list will have $\kappa'_k \leq \kappa_{max}$. □

3.5.1 Algorithm

Algorithm 1 provides steps for the proposed work. Step 6 sorts all participants in decreasing order of their connectivity. The connectivity index is added as an additional column in $P_{t_1t_2}$ matrix. So, we actually sort $P_{t_1t_2}$ on the basis of last column entry. In Step 8, upper bound is initialized to local connectivity of first node in the list. Step 11 to 15 searches for a participant with highest κ'. In Step 16, energy of the system is calculated if node having highest local connectivity is chosen as the broadcaster, and $E_{t_1t_2}$ is updated in step 18 if it minimizes energy and is included in broadcaster set.

Algorithm 1 Sorting Based Broadcaster Selection (SBS)

Input: Set of all participants: M,
Current distance matrix with $d_{ij}, \forall i, j \epsilon M$: D_t
Distance threshold when r is the localization accuracy: a_r
Moving speed of pedestrian: v
Output: Set of broadcasters for the localization period: $B_{t_1 t_2}$

Initialize $E_{t_1 t_2}$;
Calculate $P_{t_1 t_2}$;
$B_{t_1 t_2} = \phi$;
$M' = |M|$;
Initialize br;
Sort all participants in the decreasing order of their connectivity (κ_i) in $P_{t_1 t_2}$
while $M' > 0$ **do**
    $\kappa_{max} = \kappa'_1$;
    $ms = 1$;
    $j = 2$;
    while $\kappa_j > \kappa_{max}$ **do**
        if $\kappa'_j > \kappa_{max}$ **then**
            $\kappa_{max} = \kappa'_j$;
            $ms = j$;
        end if
        $j = j + 1$;
    end while
    Calculate E_{ms};
    if $E_{ms} \leq E_{t_1 t_2}$ **then**
        $E_{t_1 t_2} = E_{ms}$;
        Update br;
        $B_{t_1 t_2} = B_{t_1 t_2} \cup ms$;
        Remove ms from $P_{t_1 t_2}$;
        $M' = M' - 1$;
    else
        break;
    end if
end while

3.6 Efficiency Analysis

The proposed algorithm is a heuristic relying on connectivity and upper bound for the selection of broadcaster set. Hence, the efficiency of the algorithm largely depends on how much search space gets reduced in every iteration with the upper bound. For randomly generated instances and real dataset, the algorithm is able to minimize 12-25% of the time for medium and large datasets. However, in the worst case it is possible that search space is not reduced and entire set of participants need to be checked in every iteration for the role of broadcaster. Such a case occurs when data set is very small and connectivity of participants is low. We observe that for real datasets, our proposed algorithm terminates early.

3.7 Experimental Results

In this section, we evaluate the effectiveness of our proposed work using both real and synthetic datasets. We use SBS for denoting sorting-based broadcaster set selection algorithm and GBS for the existing greedy-based broadcaster set selection algorithm. All experiments are done over a fixed time interval, $[t_1, t_2]$ and implemented in Matlab. For all simulations, we set parameters with values according to [19][27][28], provided in Table 3.2.

Table 3.2: Parameters used for Experiments

Parameter	Values
e_g	$355mW$
e_c	$268mW$
e_{w1}	$240mW$
e_{w2}	$50mW$
l/t_c	20
l/t'_c	20
a_r	$10m = 32.8084ft$
v	$4.95ft$
$t2 - t1$	5 sec

3.7.1 Real Datasets

We evaluate the performance on GeoLife (Microsoft Research Asia) dataset [37]. This is a GPS trajectory dataset represented by sequence of time stamped points with information of latitude, longitude, and altitude. We extract four datasets from a region of high, low and medium density of size 100, 239, 320 and 880. A summary of this is provided in Table 3.3. Each participant is basically a trajectory. We have used WiFi range of 32 m for smartphone to obtain physical connectivity matrix (that is, $p_{ij} = 1$ if j is within WiFi range of node i).

Table 3.3: Dataset Summary

Dataset	**Size**	**Area**
1	100	200m× 400m
2	239	400m× 400m
3	320	2000m× 3500m
4	880	1200m× 1000m

The graph in Figure 3.6, show the energy obtained by running SBS and GBS algorithms. It is shown that there is negligible difference between the two. The slight difference occurs because of the ordering of the nodes. Two nodes can have same connectivity but physically connected to different nodes. This influences the selection of broadcasters.

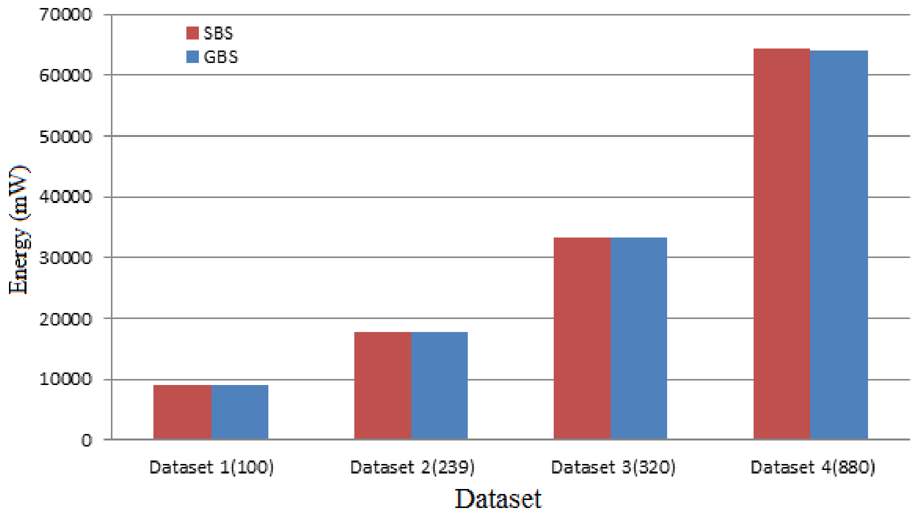

Figure 3.6: Energy Consumption by GBS and SBS Algorithms

In Figure 3.7, we depict the time taken by SBS and GBS algorithms to assign roles to the participants. It can be observed that for small datasets (for instance, size = 100) SBS and GBS takes almost same time. But for higher datasets SBS takes much less time than GBS. In real scenario, the data size is quite large. So, using SBS is more favourable than GBS.

Figure 3.7: Time taken by GBS and SBS Algorithms

3.7.2 Synthetic Datasets

We generate synthetic datasets of different sizes $\{50, 100, 200, 300, 400, 500\}$. For each, we generate random positions in terms of x and y coordinates confined in an area $500 \times 500\ m^2$. Using this we calculate distance between every node and generate a physical connectivity matrix $P_{t_1 t_2}$. Each result was obtained as an average of 20 runs.

Figure 3.8, depicts the energy difference of system obtained by subtracting power consumption when using SBS with proposed energy model from the existing model. Similarly, Figure 3.9 depict the difference when GBS is used. It evident from two graphs that modified energy model is capable of minimizing power consumption of devices and for the system. Also, difference between the two grows linearly with the size of the dataset.

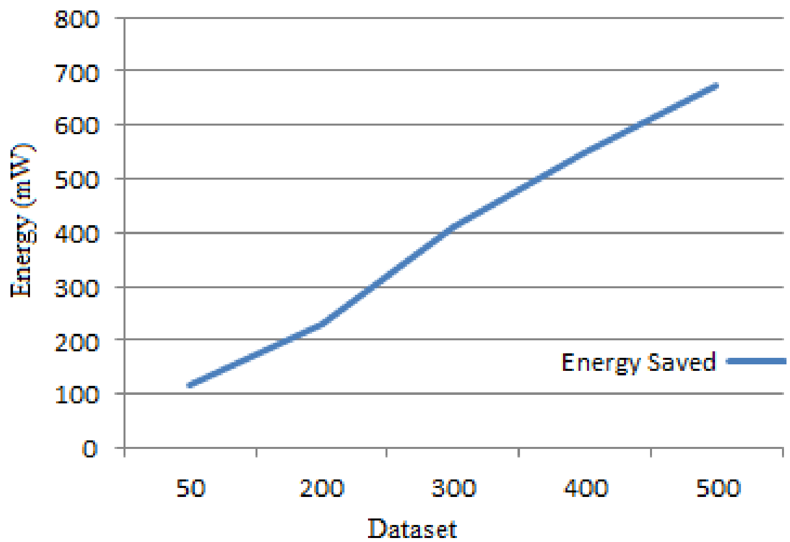

Figure 3.8: Energy Difference for SBS using Old and New Energy Model

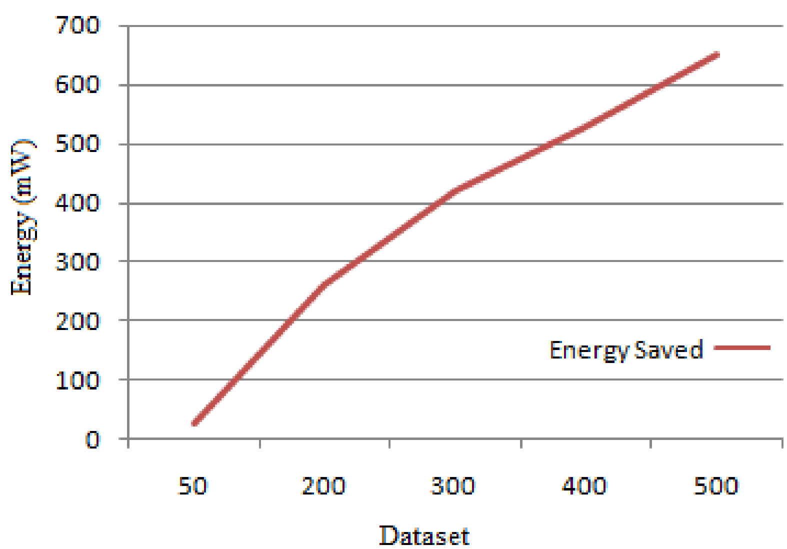

Figure 3.9: Energy Difference for GBS using Old and New Energy Model

In Figure 3.10, we once again depict the time taken by GBS and SBS algorithms for synthetic dataset. It can be observed that for dataset of size = 50, SBS takes more time than GBS. But for higher datasets (size $\geq$ 100) SBS is taking less time than GBS.

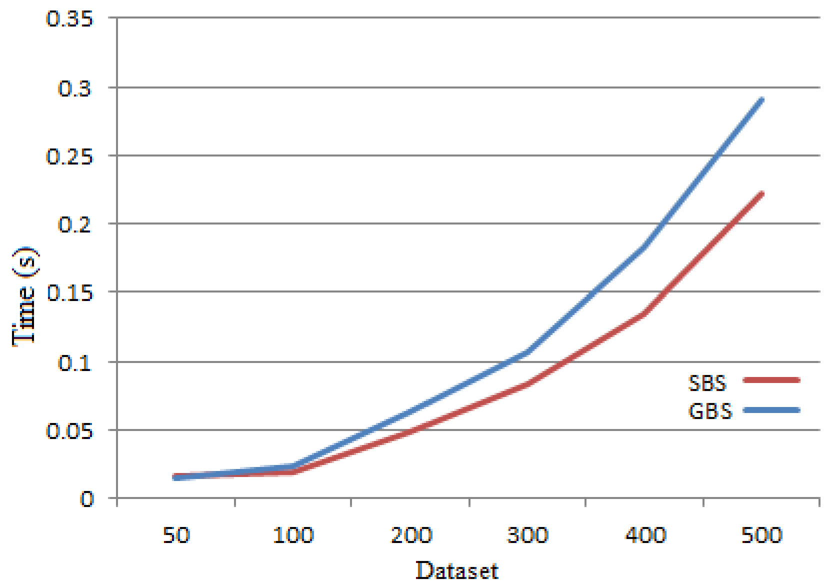

Figure 3.10: Time taken by GBS and SBS Algorithms

In Figure 3.11, we depict the power consumption when GBS and SBS algorithms were used for the role assignment. For both, we used our proposed energy model. From the graph, it is evident that the two consumes same energy. The optimal set of broadcasters found by the two algorithms is almost same. The small difference in broadcaster set size occurs because of the ordering of participants. The node chosen for the broadcaster influences the selection of next.

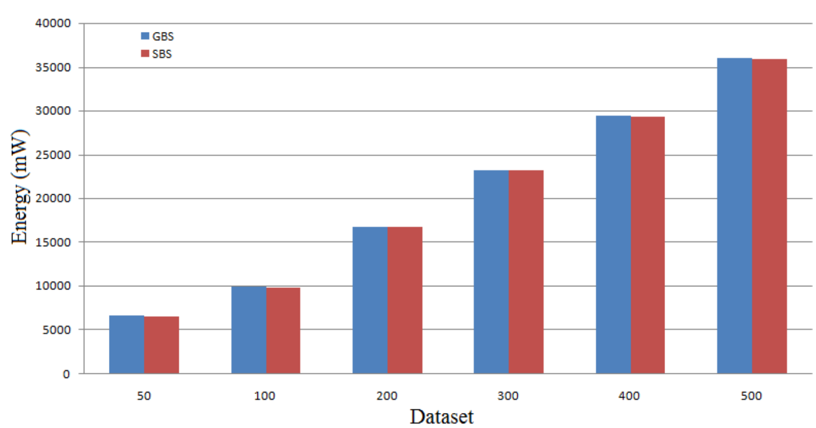

Figure 3.11: Time taken by GBS and SBS Algorithms

3.8 Conclusion

We provide enhancement to the existing work by suggesting modifications to the energy model which is capable of saving energy of the system. Also, propose a sorting based algorithm for broadcaster set selection problem. This proves to be better for data sets of moderate and higher data size which is the scenario in real cases. Experiments show that SBS takes 12-25% less time when data sets are of large size and comparable time when number of participants are small. There seems to be negligible difference in energy consumption on application of GBS and SBS algorithms which occur due to the effect of ordering of nodes.

Chapter 4

Efficient Adaptive Role Assignment for Participatory Sensing System

In previous Chapter, we have discussed centralized localization approach to detect a set of devices for the role of broadcaster which must turn on GPS. Its neighbouring devices rely on them to calculate their position. However, when the new devices join the region they cannot participate in the ongoing sensing activity. They have to wait for the next period of localization to get a role assigned to them. Similarly, if a device leaves the region then its impact on others is not yet studied.

We provide incremental algorithms to assign role to new participants entering the region and modify roles of existing devices due to exit of some participants. This is important since devices in context are mobile. The role assignment for the new device in the existing approach requires algorithm to rerun over entire set of participants. Similarly, if a device leaves the region and its neighbouring devices need to change role to minimize energy needs. Our focus is to provide a solution to eliminate the need of rerunning the algorithm during every insertion and deletion in a time efficient way without compensating energy needs.

4.1 Incremental Operations

In this approach, two operations are considered:

- *Insertion* - new participant joins the region of interest
- *Deletion* - existing participant leaves region of interest

The role of new mobile device joining the region is dependent on its location. For instance, if the device is not within WiFi range of any other device then it acts as normal participant. Similarly, the change of role for the existing devices due to deletion is affected only within the WiFi reception range of the device being deleted. Table 4.1 provides list of notations used in this Chapter with their meaning.

Table 4.1: List of notations

Notation	**Explanation**
M	Set of smartphones
$B_{t_1t_2}$	Set of broadcasters during $[t_1, t_2]$
b_m	Boolean to indicate if smartphone m is selected as broadcaster
br	Boolean to indicate if smartphone m is selected as LIR
$BLIR_{t_1t_2}$	Set of LIRs for each broadcaster for the interval $[t_1, t_2]$
κ	Connectivity of participant
κ'	Local Connectivity of a participant
$P_{t_1t_2}$	Physical connectivity matrix for interval $[t_1, t_2]$
P_{local}	Local physical connectivity matrix
I	Set of new participants being inserted
D	Set of participants to be deleted
e_b	Energy of broadcaster
e_n	Energy of normal participant
e_l	Energy of LIR
$E_{t_1t_2}$	Energy consumed during $[t_1, t_2]$
δ	Threshold for the change in database (in %)

In the following sections we discuss various cases due to insertion and deletion of a participant. Subsequently provide an algorithm for each of the approaches.

4.2 Insertion

When a participant p joins the region, new connections may be established, but none are removed. Following cases can occur:

- *(Noise)*

  If the new participant cannot become part of any broadcaster that is, it is not close enough to any broadcaster to receive its WiFi signal, then it becomes normal participant. Figure 4.1 (case 1) depicts similar case. A node x labelled as p joins an existing set of devices. However, its WiFi range does not cover any node, hence it is assigned a role of normal participant.

- *(Creation)*

  If the new participant's reception range covers few normal nodes then it is eligible to become broadcaster. It can then switch on its GPS, collect sensing information and update server through cellular network before task deadline. Figure 4.1 (case 2) illustrates this case. The green dots are used to depict normal nodes. When a new node p covers two normal nodes within its WiFi range, it is assigned a role of broadcaster and the covered nodes are reassigned the role of LIR.

- *(Absorption)*

  If the new participant is close enough to any of the existing broadcasters then it becomes part of it and act as LIR node. Figure 4.1 (case 3) shows absorption of newly inserted node, p. It is close enough to send and receive signals from an existing broadcaster depicted by red colour. Hence, gets absorbed and is assigned a role of LIR.

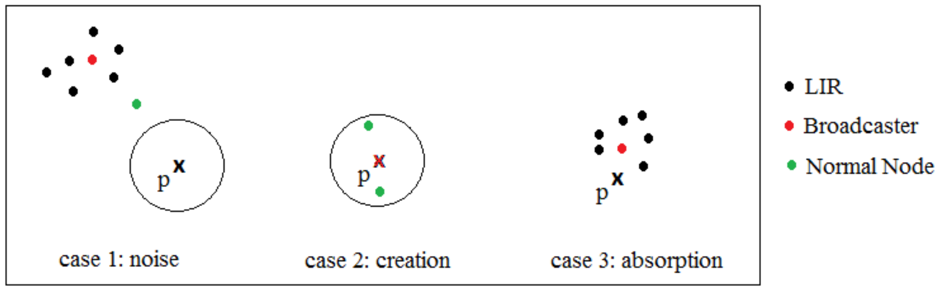

Figure 4.1: Different cases of the Insertion algorithm

4.2.1 Incremental Insertion Algorithm

In this subsection we provide algorithm for incremental insertion to consider the cases discussed above. The for loop from Step 1 to 19 iterates through entire set of participants joining the existing set of devices. Each of the parameter, E_1, E_2, and E_3 denotes energy consumed if participant is selected as broadcaster (Step 3), LIR (Step 4) or NP (Step 5) respectively. E_1 is computed using the local connectivity of new participant that is, number of normal participants that it can cover (Step 3). E_2 is updated when the participant is within WiFi range of any broadcaster. The flag is then set to 1 and energy of LIR (e_l) is added to E_2 (Step 6 to 9). E_3 is obtained by simply adding energy of NP (e_n) in case node is selected as NP. The minimum among the three energies is used to assign role to the new node (Step 11 to 17). In case new node is assigned the role of broadcaster then, the normal participants that it covers change the role to LIR and Boolean vector (br) is updated. If $br(i)$ is set to unity then it represents that i^{th} participant is assigned a role of LIR and if it is set to zero then participant can be broadcaster or normal participant.

Algorithm 2 Incremental Insertion Algorithm

Input: Initial set of participants: M,
Set of broadcaster: $B_{t_1t_2}$,
Energy $E_{t_1t_2}$ for M,
Boolean vector: br,
Set of new participants being inserted, I
Output: Role for each node in I

for each $x \in I$ **do**
    Init $flag$;
    $E_1 = E_{t_1t_2} + e_b + \kappa'_x \times (e_l - e_n)$;
    $E_2 = E_{t_1t_2}$;
    $E_3 = E_{t_1t_2} + e_n$;
    if x is physically connected to any $b \in B_{t_1t_2}$ **then**
        $flag = 1$;
        $E_2 = E_2 + e_l$;
    end if
    if $flag == 1$ && $E_2 \leq E_1$ **then**
        Assign x the role of LIR;
    else if $((flag == 1$ && $E_1 \leq E_2$ && $E_1 \leq E_3) \parallel (flag == 0$ && $E_1 \leq E_3))$ **then**
        Assign x the role of Broadcaster;
        Update br;
    else
        Assign x role of NP;
    end if
    Update $E_{t_1t_2}$;
end for

4.3 Deletion

As opposed to insertion, when a participant p leaves the region, connections or role of existing devices might change. The trickiest case is when a broadcaster is deleted. We discuss all possible cases below when deleting a node x, labelled as p:

- *(Removal)*

  If LIR leaves such that its broadcaster still covers a set of nodes then it simply updates its broadcaster with sensing data collected so far and asks to remove it from its database. LIR is removed without affecting roles of any other device. Figure 4.2 (case 1.1) shows a similar scenario. Deleting node p doesn't affect role of any other node. The broadcaster continues to cover large set of devices (LIR).

  Similarly, if normal node decides to go out of the region of interest, then it gets deleted without affecting any other participant. When server doesn't listen any update then it is considered to be deleted and database is updated. Figure 4.2 (case 1.2), shows a similar scenario. The node p is initially assigned a role of normal participant (coloured green to depict its role). However, its exit from the region doesn't affect role of any other device.

- *(Reduction)*

  Once again we consider the case when LIR leaves the region. We discussed previously how deleting LIR doesn't affect role of any other participant. The main idea for the role assignment is to minimize energy needs. If reassigning roles minimize energy more, then it is a preferred choice. For instance, in Figure 4.2 (case 2), when node p gets deleted, it leaves broadcaster with no LIR node. In such case, it is better to reassign role to this broadcaster as normal participant because $e_b > e_n$.

- *(Deletion)*

  In this we consider the case when broadcaster leaves the region. Since it knows the location of every neighbouring participant (its LIR devices) so it runs sorting based or greedy algorithm over its LIRs and reassign roles to them. New broadcasters are chosen if they can cover LIRs otherwise node reassign role of normal participant. Figure 4.2 (case 3.1), depicts the case when a node can be chosen as broadcaster which covers all LIRs of p. However in Figure 4.2 (case 3.2), when a broadcaster node p decides to leave the region, it leads to formation of two other broadcasters to cover its LIR nodes.

  The broadcaster being deleted shares its data, task and deadline with newly assigned broadcasters and normal participants. The information of the node being deleted would be updated to the server at the end of the task.

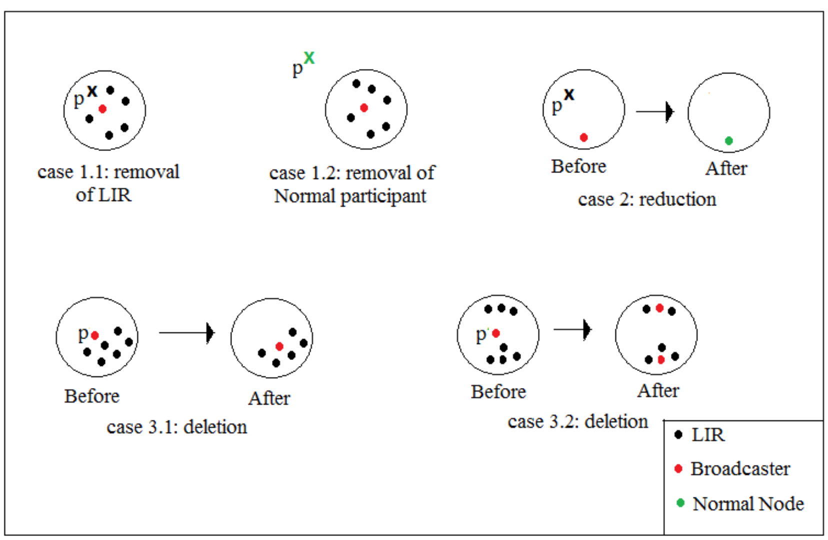

Figure 4.2: Different cases of the Deletion algorithm

4.3.1 Incremental Deletion Algorithm

In this subsection we provide an algorithm for incremental deletion (Algorithm 3) to consider all the cases discussed above. For each deletion (Step 1), we find their existing role. If this participant x, happens to be a NP then it is simply removed (Step 39). If x was assigned a role of LIR then we remove it from LIR set by replacing one with zero in br vector (Step 3). We find its corresponding broadcaster using $BLIR_{t_1t_2}$ matrix(Step 4). The $BLIR_{t_1t_2}$ maintains the LIRs that broadcaster covers. Whenever a device is chosen as broadcaster, its index is appended in matrix $B_{t_1t_2}$. Its corresponding set of LIRs represented by Boolean vector, is appended in Boolean matrix, $BLIR_{t_1t_2}$. The bit at (i, j) is set to unity when j^{th} device is chosen as LIR for i^{th} broadcaster.

In case the broadcaster of x no longer cover any LIR then its role is changed to normal participant (Step 6 to 9). If participant, x was assigned the role of broadcaster then we obtain a local physical connectivity matrix, P_{local} (Step 12). Our aim is to find new set of broadcasters that can cover its LIRs. For this, every entry of P_{local} is obtained by the operation of Boolean AND over LIR set of x and physical connectivity of its LIRs (Step 13 to 17). Next we obtain local connectivity of x, represented by κ'_x (Step 18). We sort P_{local} in the descending order of participant's connectivity (Step 19). We then select participants from P_{local} that can cover LIRs of x (Step 30 to 35). The local connectivity of each participant is obtained by the Boolean AND operation (Step 24). In case they do not cover any LIR then they are removed from LIR set (Step 26) and assigned role of normal participant (Step 27). The local connectivity of x is decremented by 1 whenever its LIR is assigned a role of normal participant (Step 28). If participant covers some LIRs of x then it is added to broadcaster set (Step 31) and corresponding LIR set is added to $BLIR_{t_1t_2}$ (Step 32). With this we update the local connectivity of x by removing participants that have been covered using Boolean XOR operation (Step

33) and removing the participant from $x's$ LIR list that is recently chosen for broadcaster role(Step 34) and global LIR list, br. The κ'_x is then updated (Step 35).

4.4 Proposed Model

In dynamic scenario we consider participants joining or leaving the existing set of devices on fly and aim to assign role without rerunning greedy or sorting based algorithms for each insertion or deletion. However, our aim is to minimize energy. The proposed incremental algorithms provided in Algorithm 2 and 3 does not provide optimal set of broadcasters. Hence, there is a need to rerun the algorithm when energy consumption becomes too high. We set a threshold, δ to check the change in database. If change in database is less than δ then it is better to use incremental technique to assign role otherwise we advise to rerun SBS or GBS algorithm.

The percentage change in database can be calculated by following:
$$\% \text{ Change in DB} = \frac{NewData - OldData}{OldData} \times 100.$$

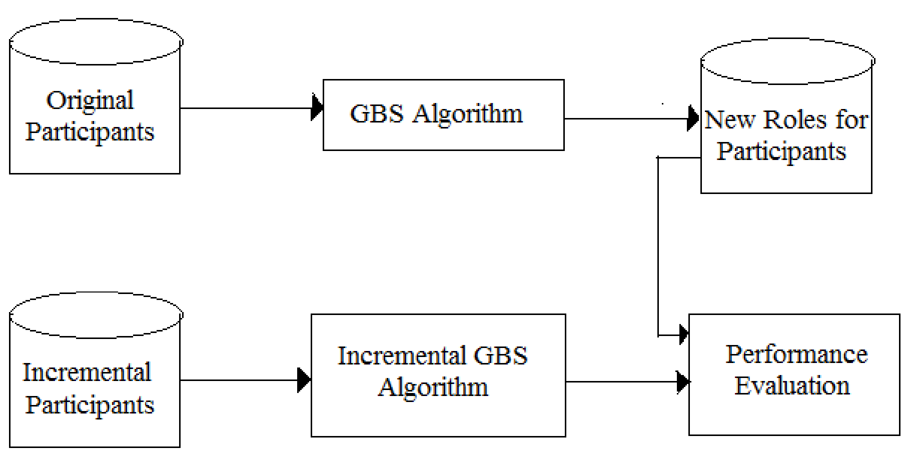

Figure 4.3: Proposed Model for Adaptive Algorithm

The proposed model is depicted by Figure 4.3 and its pseudocode is provided in

Algorithm 3 Incremental Deletion Algorithm

Input: Initial set of participants: M, Set of broadcaster: $B_{t_1t_2}$, Boolean vector: br, Set of LIRs for each broadcaster: $BLIR_{t_1t_2}$, Physical connectivity matrix: $P_{t_1t_2}$, Energy $E_{t_1t_2}$ for M, Set of participants to be deleted, D

Output: Role for each node in M

for each $x \in D$ **do**
    if $br(x)$==1 **then**
        $br(x) = 0;$
        $b = x's$ broadcaster;
        $BLIR_{t_1t_2}(b, x) = 0;$
        if $sum(BLIR_{t_1t_2}(b)) == 0$ **then**
            b becomes NP;
            Remove b from $B_{t_1t_2}$;
        end if
    else if x exists in $B_{t_1t_2}$ **then**
        i=index of x in $B_{t_1t_2}$;
        Initialize P_{local};
        for each $j : BLIR_{t_1t_2}(i, j) = 1$ **do**
            $temp$=$BLIR_{t_1t_2}(i) \wedge P_{t_1t_2}(x)$;
            $temp(j)$=0;
            Insert $temp$ in P_{local};
        end for
        $\kappa'_x = sum(BLIR_{t_1t_2}(i));$
        Sort P_{local} in descending order of κ;
        $iter$ = number of rows in P_{local};
        $k = 1;$
        while $\rho\ = 0$ && $k\ = iter$ **do**
            m =Index of $P_{local}(k)$;
            $\kappa' = sum(BLIRt_1t_2(i) \wedge P_{local}(k);$
            if $\kappa' == 0$ **then**
                $br(m) = 0;$
                Assign role of NP to m;
                $\kappa'_x = \kappa'_x - 1;$
            else
                $br(m) = 0;$
                $B_{t_1t_2} \cup \{m\};$
                $BLIR_{t_1t_2} = BLIR_{t_1t_2} \cup P_{local}(k);$
                $BLIR_{t_1t_2}(i) = BLIR_{t_1t_2}(i) \otimes P_{local}(k);$
                $BLIR_{t_1t_2}(i, m) = 0;$
                $\kappa'_x = \kappa'_x - \kappa' - 1;$
            end if
        end while
    else
        Remove x from NP set;
    end if
end for

Algorithm 4. The actual SBS (or GBS) algorithm is applied to the original database to assign role to each participant (Step 1). Then we use incremental insertion and deletion algorithm to adapt new changes to the dataset if the change is less than threshold, δ(Step 3,4) otherwise rerun SBS/GBS (Step 6).

Algorithm 4 Adaptive Algorithm

1: Apply SBS/GBS on the original set of participants.
2: **for** each x inserted or deleted **do**
3:     **if** % change in database $< \delta$ **then**
4:         Apply incremental insertion or deletion algorithm to assign role to x
5:     **else**
6:         Rerun SBS/GBS algorithm
7:     **end if**
8: **end for**

4.5 Performance Evaluation

We evaluate the performance of proposed adaptive strategy using synthetic dataset of size 100, 300, 500 and 600. For each participant, we generate random positions in terms of x and y coordinate confined in area of $500 \times 500\ m^2$. A distance matrix is generated to calculate distance between every device and then derive physical connectivity matrix. All parameters are set with same values as used in [19]. We have used Matlab for all simulations and experiments. We have not experimented with GBS algorithm as its performance is already evaluated in previous Chapter and the results remain same if it is compared with incremental GBS. In first sub section, we discuss results of insertion algorithm which is followed with the results of deletion algorithm in next.

4.5.1 Incremental Insertion

In each of the results, we call the proposed algorithm as *Incremental Insertion* and *SBS* for rerunning SBS algorithm for each insertion. Each of the following results were obtained as an average of 25 runs on 25 datasets each of the size {100, 300, 500, 600}.

In the first experiment, we aim to evaluate the time taken for assigning roles by proposed incremental insertion and SBS algorithm. For this the number of new participants inserted is equivalent to 5% of the dataset size. The sorting based algorithm has to rerun for every insertion. Figure 4.4 clearly depicts that our algorithm outperforms SBS.

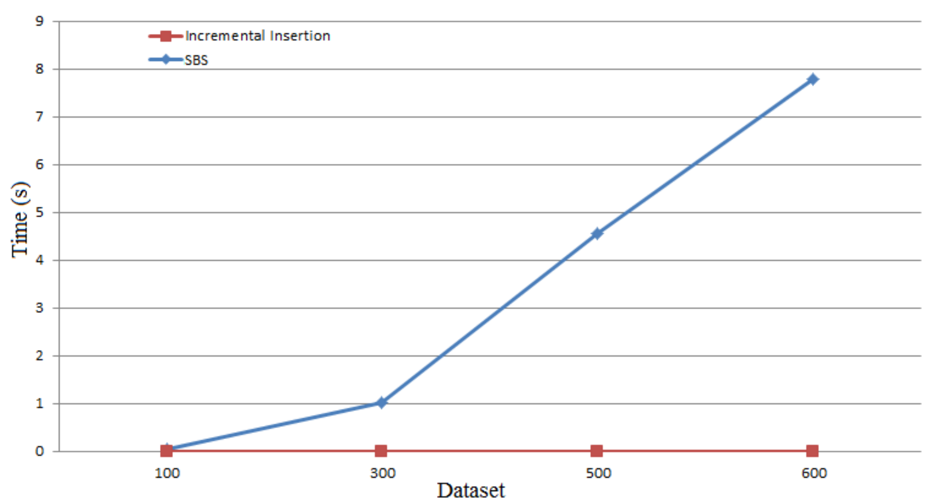

Figure 4.4: Time taken by SBS and Incremental Insertion Algorithms for role assignment

In the next experiment, we evaluate the impact on energy consumption as a result of role assignment using the two algorithms. This is essential as SBS algorithm finds an optimal set of broadcaster so consumes minimum energy. Our aim is to check whether the new approach is efficient enough to assign roles such that it does not consume too much of energy. In this experiment also, we added new participants equivalent 5% of the data set size. Figure 4.5 shows system energy consumed when roles are assigned by

SBS and incremental insertion approach. It can be observed that energy consumed by the proposed algorithm is almost equivalent to that of the optimal algorithm.

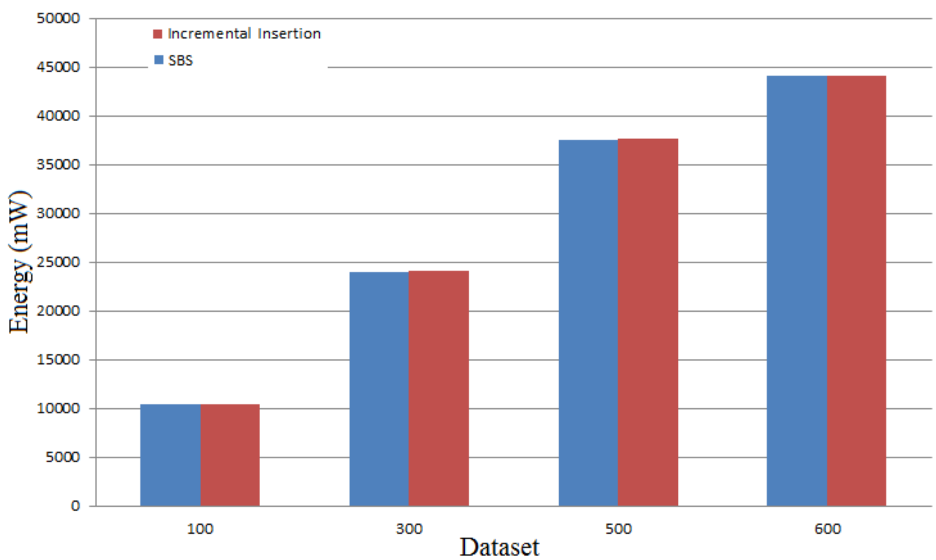

Figure 4.5: Energy Consumption by SBS and Incremental Insertion Algorithms

Next, we evaluate the nature of processing time on a dataset of size 300 as we incrementally insert 15 participants. From Figure 4.6, we observe that the rate of processing remains same for every insertion in proposed algorithm. However, there is slight variation for SBS algorithm which happens when an inserted node increases or decreases number of broadcasters. We also observe that time taken by SBS algorithm steadily increases. This is so because the size of the dataset increases with every insertion.

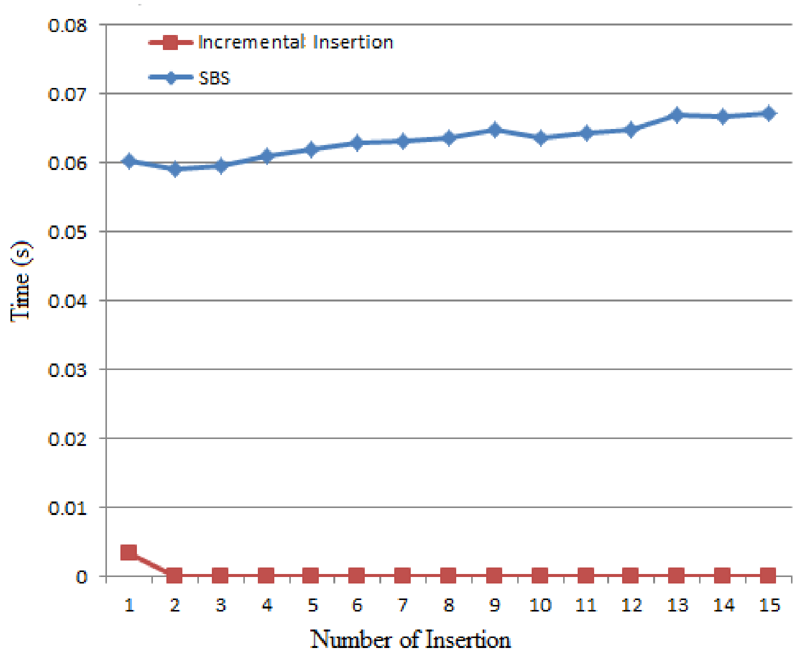

Figure 4.6: Time versus Number of Insertions

In the last set of experiment, we analyse a threshold after which the SBS algorithm must be repeated. This is evaluated by varying percentage of insertion and observing the difference in energy between the incremental and SBS algorithms. For this experiment, we consider dataset of size 100, and incrementally insert 5%, 10%, 15% and 20% nodes. From Figure 4.7, we can observe that when number of insertions is greater than 10% then incremental consumes more energy. So, 10% to 15% can be selected as a threshold for rerunning SBS algorithm.

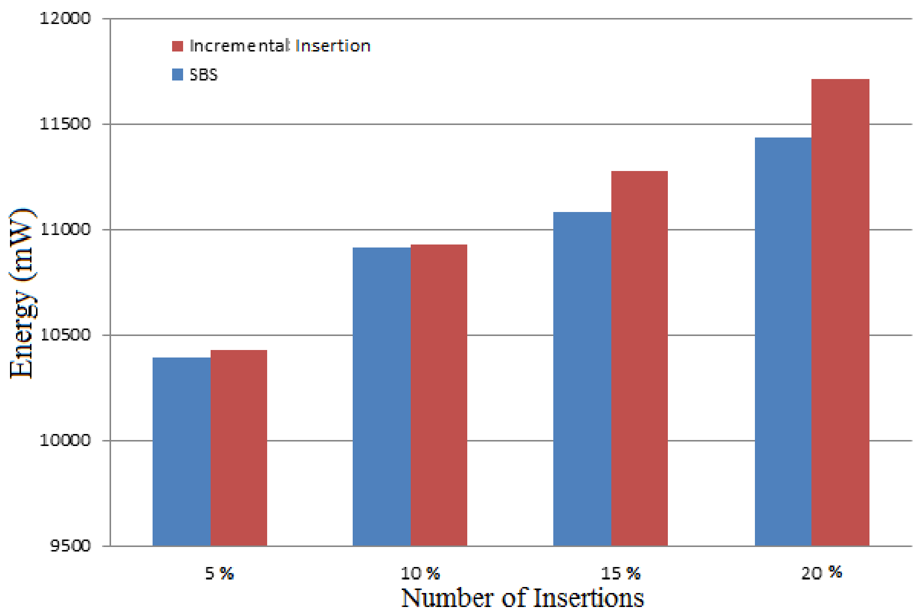

Figure 4.7: Energy Consumption for {5%, 10%, 15%, 20%} Insertions

4.5.2 Incremental Deletion

In each of the results, we call the proposed algorithm as *Incremental Deletion* and *SBS* for rerunning SBS algorithm. Each of the following results was obtained as an average of 25 runs for each dataset of size {100, 300, 500, 600}. For this we consider one data set each of the size mentioned above and generated 25 sets of uniformly distributed random numbers. These provided indices of participants which were used to delete nodes.

In the first experiment, we evaluate time taken by proposed incremental deletion and SBS algorithm for role assignment. The number of participants which were deleted was equivalent to 5% of the dataset size. The sorting based algorithm had to rerun for every deletion. Figure 4.8 clearly depicts that the proposed algorithm takes much less time than SBS.

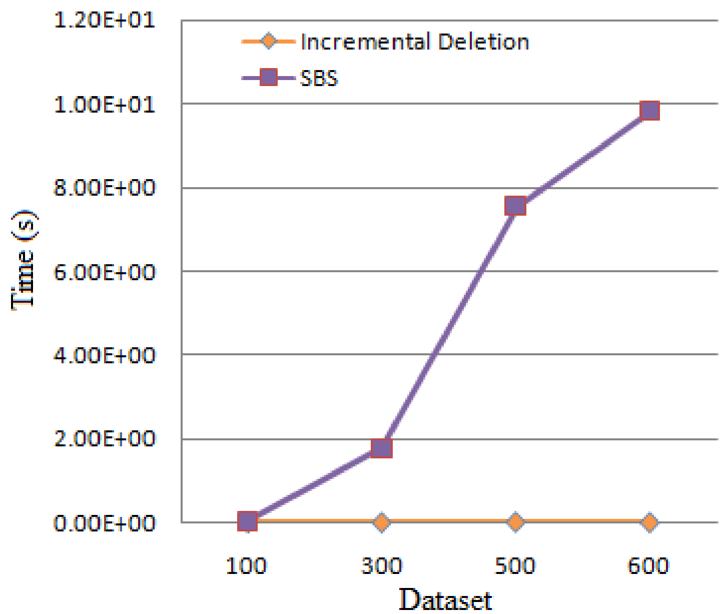

Figure 4.8: Time taken by SBS and Incremental Deletion Algorithms for role assignment

Next, we evaluate the impact on energy consumption as a result of role assignment using the two algorithms. This is required to evaluate performance of proposed algorithm over the optimal algorithm as energy is vital in mobile sensing. In this experiment we deleted 5% of the total participants in consideration. Figure 4.9 shows system energy consumed when roles are assigned by SBS and incremental deletion approach. It can be observed that energy consumed by the proposed algorithm is almost equivalent to that of the optimal algorithm.

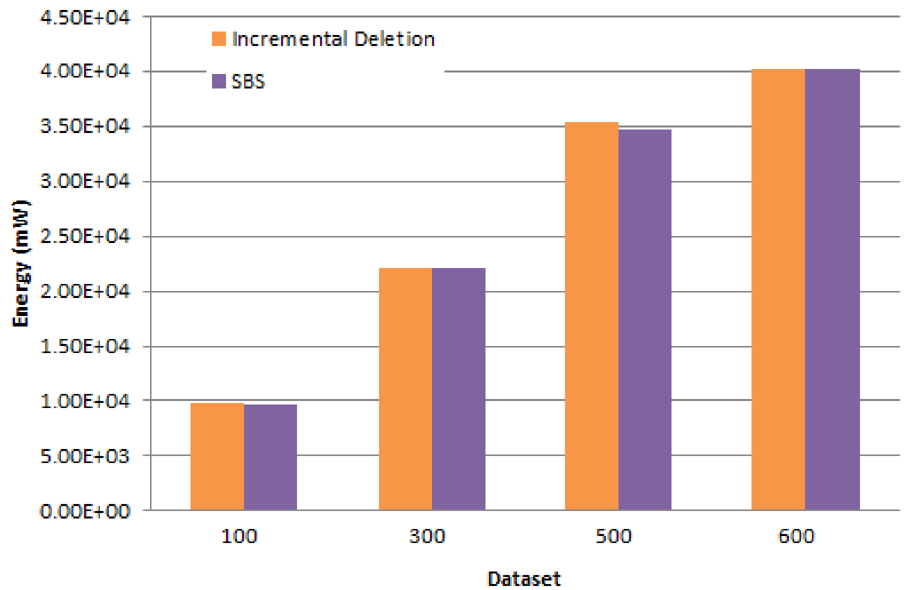

Figure 4.9: Energy Consumption by SBS and Incremental Deletion Algorithms

In next experiment, we evaluate the nature of processing time on a dataset of size 300 as we incrementally delete 15 participants. From Figure 4.10, we observe that the rate of processing remains same for every deletion in proposed algorithm. However, the variation in SBS algorithm occurs due to the impact of deletion on number of broadcasters. We also observe that in contrast to insertion, the time taken by SBS algorithm steadily decreases. This is so because the size of the dataset decreases with every deletion.

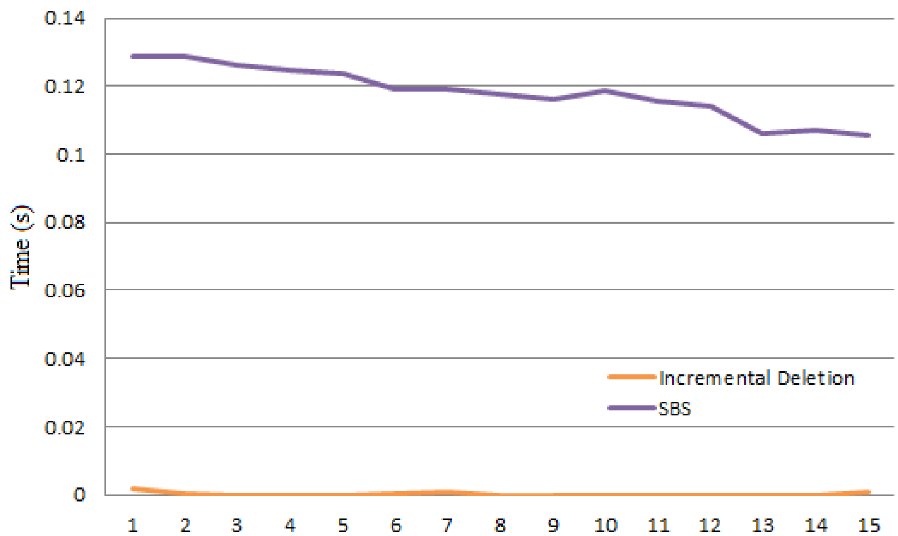

Figure 4.10: Time versus Number of Deletion

In the last set of experiment, we analyse a threshold after which the SBS algorithm must be repeated. This is evaluated by varying percentage of deletion and observing the difference in energy between proposed and optimal algorithm. For this experiment, we consider dataset of size 100, and incrementally delete 5%, 10%, 15% and 20% nodes. From Figure 4.11, we can observe that when number of deletions is greater than 5% then incremental consumes more energy. So, a threshold between 10-15% can be chosen for rerunning SBS algorithm.

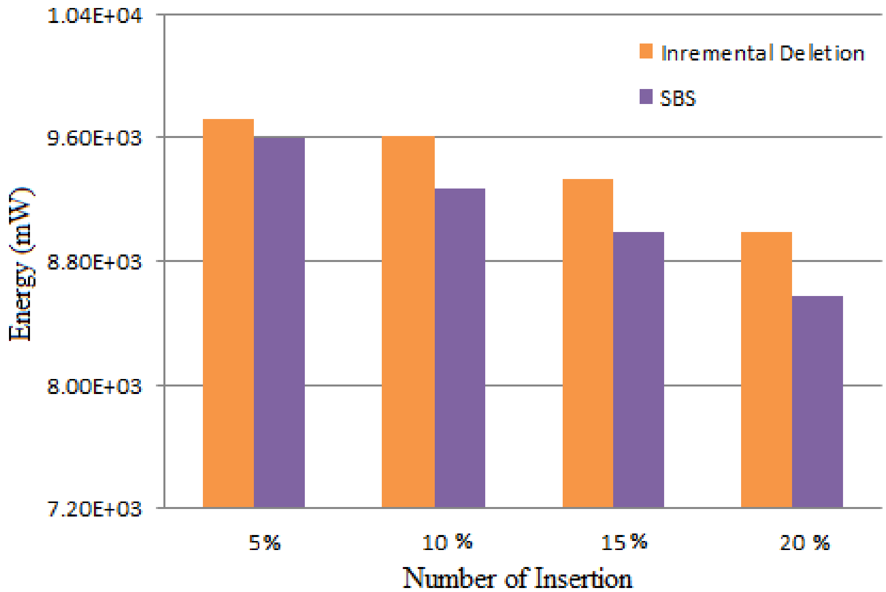

Figure 4.11: Energy Consumption for {5%, 10%, 15%, 20%} Deletions

4.6 Conclusion

The existing techniques based on greedy and sorting based approaches are not efficient to consider the adaptive changes to the dataset. Only way to assign role to new participant or change roles of existing devices due to deletion of some device is to rerun the algorithm. But as the data set can be large enough so it is not worth running algorithm for each change. In lieu of that concern, we propose incremental algorithms that helps saving time and effort. We provided approaches for insertion and deletion of participants on fly and their corresponding algorithms. From experimental results we can see that proposed solutions saves 95-99.9% of time for role assignment compared to SBS algorithm. Also when change of the dataset is within [5%, 10%], the role assignment using these algorithms provide almost same energy consumption of system as obtained with optimal algorithms.

Chapter 5

Distributed Localization for Participatory Sensing System

In this Chapter, we provide a novel distributed solution for finding the roles for the participating mobile devices. In the previous two Chapters we discussed centralized approaches where the server assigns role to each device. However, in real scenario it is quite infeasible for the server to be burdened with extra work of role assignment. We have seen a localized approach used by LIRs to detect their location. We propose another level localization in which every node decides its role.

Our proposed approach does not make any assumptions about presence of infrastructure or device capabilities, other than availability of WiFi and ability to know devices within WiFi range. We present protocol distributed localization for participatory sensing system. Simulation results demonstrate the capability of finding broadcaster set in very less time compared to centralized algorithms.

5.1 System Overview

We consider a participatory sensing system shown in Figure 5.1. It consists of server (sink), task publisher and set of smartphone users in the region. The task publisher sends the sensing task to server.

In the role assignment phase, all participants decide their role and in localization period, they find their location and collect sensing data. There can be three type of roles as discussed in [19] - broadcaster (aggregators), location information receivers (LIRs), normal participants (NP).

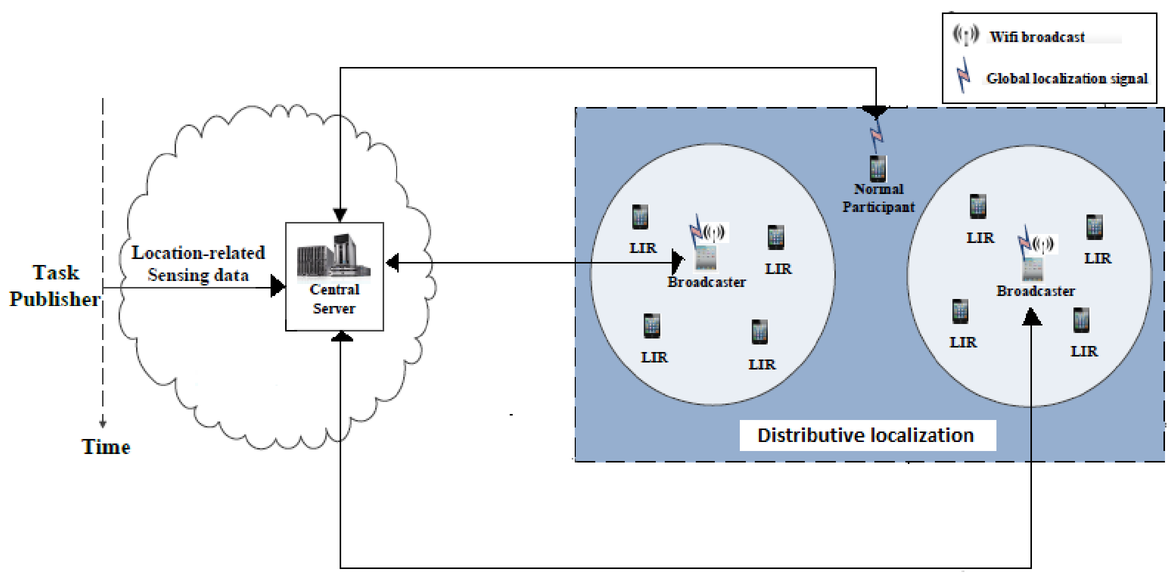

Figure 5.1: System Architure

Only broadcasters and normal participants sends back data to the server which processes it to accomplish the given task. We are considering that users sense environment irrespective of the task and server can process the aggregated information as per the requirement. However, it can be considered that sensing is performed based on task. In which case server sends back task information to broadcasters and normal participants which shares it with LIRs.

We assume following properties for the system:

- All users are mobile. This corresponds to realistic case.
- The server act as sink and processes data for the task .
- All nodes are equipped with GPS, accelerometer and other sensors which are used as per requirement.
- Nodes are also equipped with WiFi and cellular network but are used only when needed; rest of the time they are turned off.

No assumptions are made about

- homogeneity of node dispersion
- network density
- distribution of energy

5.2 Objective

We aim to achieve following requirements:

- Role assignment must not involve server and should be completely distributed. Each node must make decision independently based on the local information.
- At the end of role assignment phase, each node has assigned a part of either a broadcaster, LIR or normal participant to itself.
- The algorithm should terminate within fixed number of iterations.

5.3 Energy Consumption Model

We consider that nodes periodically execute distributed algorithm to reassign their role. Let this interval be $[t_1, t_2]$. If t_x is the time taken for role assignment then $[t_1 + t_x, t_2]$ is time taken for the localization.

We use same notations as used in Chapter 3 to denote power of GPS, cellular network, WiFi sending and receiving and energy of broadcaster, LIR, NP and entire system.

Figure 5.2, Figure 5.3 and Figure 5.4 shows energy consumption at different intervals for broadcaster, LIR and normal participants respectively during $[t_1, t_2]$. Every node keeps WiFi on until role assignment so utilize $e_{w1} + e_{w2}$ during $[t_1, t_1 + t_x]$ irrespective of whether they become broadcaster, LIR or normal participant. Rest remains same as discussed in Chapter 3.

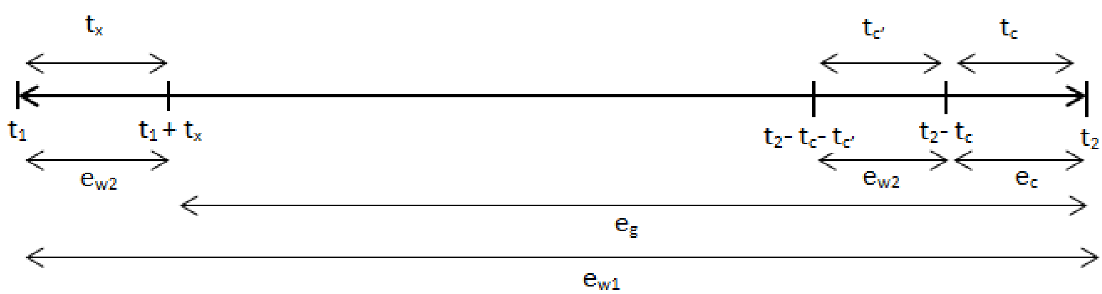

Figure 5.2: Energy Consumption of Broadcasters during $[t_1, t_2]$

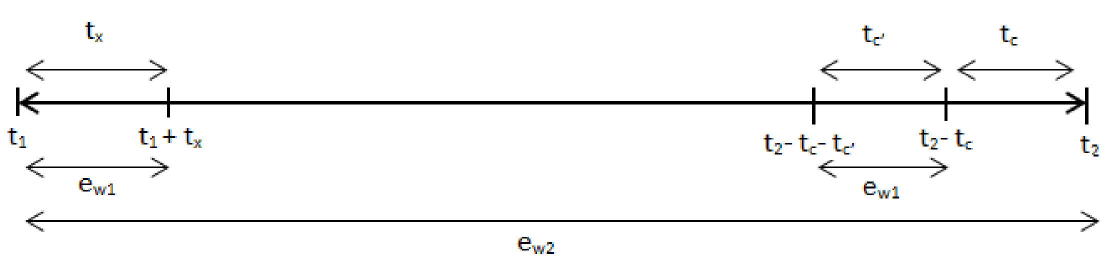

Figure 5.3: Energy Consumption of LIRs during $[t_1, t_2]$.

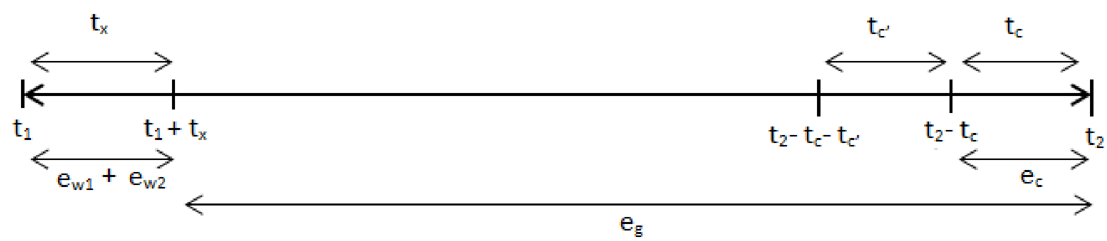

Figure 5.4: Energy Consumption of normal participants during $[t_1, t_2]$.

On the basis of Figure 5.2, Figure 5.3 and Figure 5.4, the energy for broadcasters (e_b), LIR (e_l) and NP (e_n) can be written as:

$$e_b = (e_{w2} \times (t_x + t_{c'}) + e_{w1} \times (t_2 - t_1) + e_g \times (t_2 - t_1 - t_x) + e_c \times t_c)/(t_2 - t_1) \quad (5.1)$$

$$e_l = (e_{w1} \times (t_x + t_{c'}) + e_{w2} \times (t_2 - t_1))/(t_2 - t_1) \quad (5.2)$$

$$e_n = ((e_{w1} + e_{w2}) \times t_x + e_g \times (t_2 - t_1 - t_x) + e_c \times t_c)/(t_2 - t_1) \quad (5.3)$$

5.4 The Distributed Protocol

In this section, we describe the protocol by first defining few terms. Then we also make an attempt to prove that objectives are met.

5.4.1 Terminology

Each node requires fix number of iterations to assign a role to itself which we denote as N_{iter}. From centralized schemes, we have observed that the node which is connected to most other nodes is suitable for broadcaster role. This conveys that connectivity of a node is an important factor for becoming broadcaster. However, we also consider the residual energy of nodes because devices in context are smartphones so it is possible that their energy is drained by other running applications. As a result, nodes with high connectivity and residual energy are given more priority over others to become broadcaster. So far, residual energy has not been considered even for the centralized approach where nodes are only checked for minimum energy constraint to become eligible for broadcaster. This is first work which considers both connectivity and node's residual energy. To explain the proposed algorithm, we make use of the terms - *connectivity (κ)* and *local connectivity (κ')* introduced in Chapter 3 to define few more terms.

Definition 5 *Reachability probability ($P_r(i)$) is defined as the ratio of local connectivity of node i to the total number of participants in the system. Mathematically, it can be expressed as:*

$$P_r(i) = \frac{\kappa'_i}{\#participants}$$

where $\#participants$ denote total number of participants in the system.

Definition 6 *Energy Ratio ($P_e(i)$) is defined as the ratio of residual energy in the node i to the battery capacity of smartphone. Mathematically, it can be expressed as:*

$$P_e(i) = \frac{ResidualEnergy_i}{TotalEnergy_i}$$

Definition 7 *Broadcaster probability ($P_b(i)$) of node i can be expressed as the product of iteration step and weighted sum of reachability probability and energy ratio.*

$$P_b(i) = N_{iter} \times [\alpha \times P_r(i) + (1 - \alpha) \times P_e(i)]$$

where α denotes a weight factor $\in [0, 1]$.

We can assign α a value of 0.5 to give equal priority to reachability probability and energy ratio.

5.4.2 Distributed Algorithm

Role assignment and localization phase is repeated periodically in an interval of $[t_1, t_2]$ to select next set of broadcasters. We make sure that broadcaster probability of devices does not get too low so, we set a threshold of 0.01. This helps to terminate the algorithm in $O(1)$ iterations. The use of energy ratio in the broadcaster probability helps to extend it to heterogeneous mobile devices with varying battery life as well. This is an extension to the centralized approach which does not takes into account the residual energy and therefore, treats all devices same.

In this scheme we consider that a node can be part of only one broadcaster which can be easily extended to multiple broadcasters as well. As per the assumption, every node is aware of devices within its WiFi range. If a node becomes LIR then its neighbours are aware of its new role. This follows from the observation that if node decides to become broadcaster then this information is also available to neighbours which assign role of LIR to themselves.

All devices which do not have any node within their WiFi range assign the role of normal participant (Step 6) to themselves. Remaining nodes iterate to find their role. In each iteration (Steps 11 to 24), every node reassigns its broadcaster probability (Step 13, 14) to check if they are eligible to become broadcaster. If probability becomes greater than equal to 1 then node assigns role of broadcaster and subsequently all neighbouring nodes which have not become part of any broadcaster, assigns the role of LIR (Step 15 to 20). The N_{iter} is doubled in every iteration (Step 23). This ensures that broadcaster probability is doubled in every step. The probability is recomputed to take into account the dynamic nature of mobile device. In case nodes change their location then its new connectivity can be considered without affecting the N^{th} iterations that it had to execute. This is unique to distributed algorithm as change of node's locality is also considered. The centralized approach uses the initial positions registered to the server for assigning role which remains same for entire role assignment and localization period. The while loop from Step 10 to 25 ensures that every node has assigned a role to itself.

Algorithm 5 Distributed Algorithm

Input: Set of all participants: M,
Current distance matrix with $d_{ij}, \forall i, j \epsilon M$: D_t
Output: Role for every node

1: Calculate $P_{t_1t_2}$;
2: $B_{t_1t_2} = \phi$;
3: $L_{t_1t_2} = \phi$;
4: $N_{t_1t_2} = \phi$;
5: Init br;
6: $N_{t_1t_2} \leftarrow$ All nodes with $(\kappa = 1)$;
7: $M = M - N_{t_1t_2}$;
8: $M' = |M|$;
9: $N_{iter} = 1$;
10: **while** $M' > 0$ **do**
11:     **for** every node i in M **do**
12:         **if** i is not assigned any role **then**
13:             Compute $P_b(i)$;
14:             $P_b(i) = max(P_b(i), 0.01)$;
15:             **if** $P_b \geq 1$ **then**
16:                 $B_{t_1t_2} = B_{t_1t_2} \cup \{i\}$;
17:                 $L_{temp} \leftarrow$ All nodes in M within i's WiFi range;
18:                 $M = M - L_{temp}$;
19:                 $L_{t_1t_2} = L_{t_1t_2} \cup L_{temp}$;
20:             **end if**
21:         **end if**
22:     **end for**
23:     $N_{iter} = N_{iter} \times 2$;
24:     $M' = |M|$;
25: **end while**

5.4.3 Correctness

Lemma 4 *The algorithm terminates in $N_{iter} = O(1)$ iterations.*

Proof: We consider worst case when $\alpha = 1$ and $\alpha = 0$. It can be proved for other values of α also in the same way.

Case 1: ($\alpha = 1$) The best case occurs when connectivity of node is very less (that is, 0) or very high (that is, 1). When $P_r = 0$, node assigns role of normal participant. When $P_r = 1$, node assigns role of broadcaster. In both the cases $N_{iter} = 1$. Worst case occurs when node has $0 < P_r < 1$. P_b can be minimum 0.01 which is our threshold. As this probability is doubled in every iteration until it reaches 1 hence, $2^{N_{iter}-1} \times 0.01 \geq 1$ and

$$N_{iter} \leq \lceil \log_2 \frac{1}{0.01} \rceil + 1 = 8$$

Therefore, $N_{iter} \approx O(1)$. The threshold can be modified but it will be still bounded by a reasonable constant.

Case 2: ($\alpha = 0$) The worst case occurs when a node has very low residual energy and it sets its P_b to 0.01. As shown above, the maximum number of iterations that will be carried by this node will be 8. Hence, algorithm terminates in $O(1)$ iterations. $\square$

Lemma 5 *At the end of the algorithm, each node has assigned a role to itself.*

Proof: A node which does not have any node within its WiFi range is eligible to become normal participant. In case it doesn't, then it can assign role of broadcaster or LIR. Assume that node terminates without assigning any role. This means that node doesn't belong to WiFi range of any of the broadcasters. In such a case, the broadcaster probability would reach 1 and node assigns a role of broadcaster to itself, which is a contradiction. $\square$

5.4.4 Complexity

The time complexity is least compared to sorting based (SBS) and greedy based (GBS) algorithm for the centralized approach.

Lemma 6 *The distributed algorithm has a worst case processing time complexity of $O(n)$ per node, where n is the number of nodes in the network.*

Proof: Time taken by each node is atmost N_{iter} which is a constant according to Lemma 4. So, total time will be $N_{iter} \times n = O(n)$, □

5.5 Distributed versus Centralized Approach

In this section, we tabularize the difference between distributed and centralized approach highlighting merits and demerits of the two in Table 5.1.

5.6 Performance Analysis

We evaluate the performance on GeoLife (Microsoft Research Asia) dataset [37]. This is a GPS trajectory dataset representing sequence of time stamped points with information of latitude, longitude, and altitude. The dataset comprises 17621 trajectories of 182 users collected from April 2007-August 2012. Most of it is collected in China (over 30 cities) and some parts of USA, and Europe as well.

We extract four datasets from a region of high, low and medium density of size 100, 239, 320 and 880. A summary of this is provided in Table 5.2. Each participant is basically a trajectory. We have used WiFi range of 32 m for smartphone to obtain physical connectivity matrix; $p_{ij} = 1$ if j is within WiFi range of node i.

Table 5.1: Difference between Distributed and Centralized Approach

Distributed Approach	**Centralized Approach**
Devices decide role themselves.	Roles are decided by central server.
Roles are decided based on local information.	Roles are decided based on global information.
There is no guarantee of having optimal number of broadcasters because nodes decide roles themselves.	An optimal number of broadcasters are selected as it is dependent on centralized source for role assignment.
Role assignment takes less time as compared to centralized approach as it takes $O(1)$ iterations.	Role assignment takes more time.
Size of broadcaster set is more or equal to the size found in centralized approach.	Size of broadcaster set is smaller.
Total energy dissipated is more than centralized approach.	Total energy dissipated is less than distributed approach.
It takes into account connectivity and residual energy of smartphones.	It takes into account only connectivity for role assignment.
It can handle heterogeneous nodes.	It takes into account homogeneous nodes as residual energy is not considered.

Table 5.2: Dataset Summary

Dataset	**Size**	**Area**	$\frac{\mathbf{1}}{\mathbf{Density}}$
1	100	200m × 400m	400
2	239	400m × 400m	669.344
3	320	2000m × 3500m	21881.8381
4	880	1200m × 1000m	1386.86301

Energy and other parameters were set with values used in Chapter 3. For the first three experiments, we set α to 0.5 and residual energy equivalent to the total energy of the mobile device for distributed algorithm because centralized approach does not consider residual energy of devices.

We evaluate the time taken by distributed algorithm, sorting and greedy-based broadcaster selection algorithms. Figure 5.5 shows that distributed algorithm outperforms other two taking minimal time because every node executes fixed number of iterations.

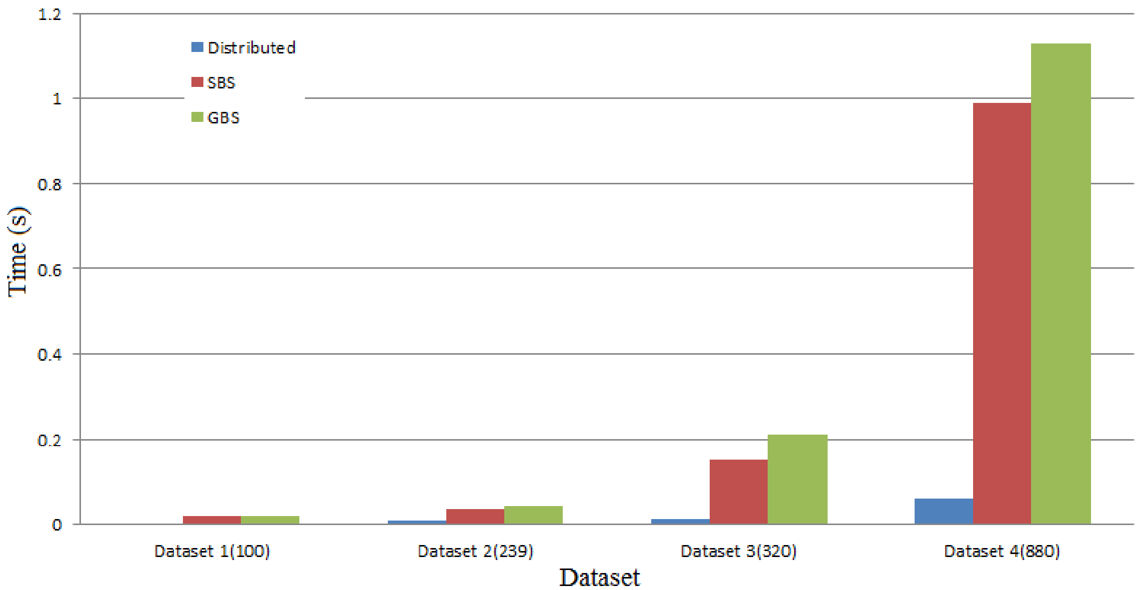

Figure 5.5: Time taken by Distributed, SBS and GBS Algorithms

In the second experiment we present the energy consumption of the system by using the three algorithms. Once again α is set to 0.5 and residual energy same as the total energy. Figure 5.6 shows that distributed algorithm consumes more energy while SBS and GBS consume the same. This is because SBS and GBS finds optimal set of broadcaster and hence consumes less energy compared to distributed scheme.

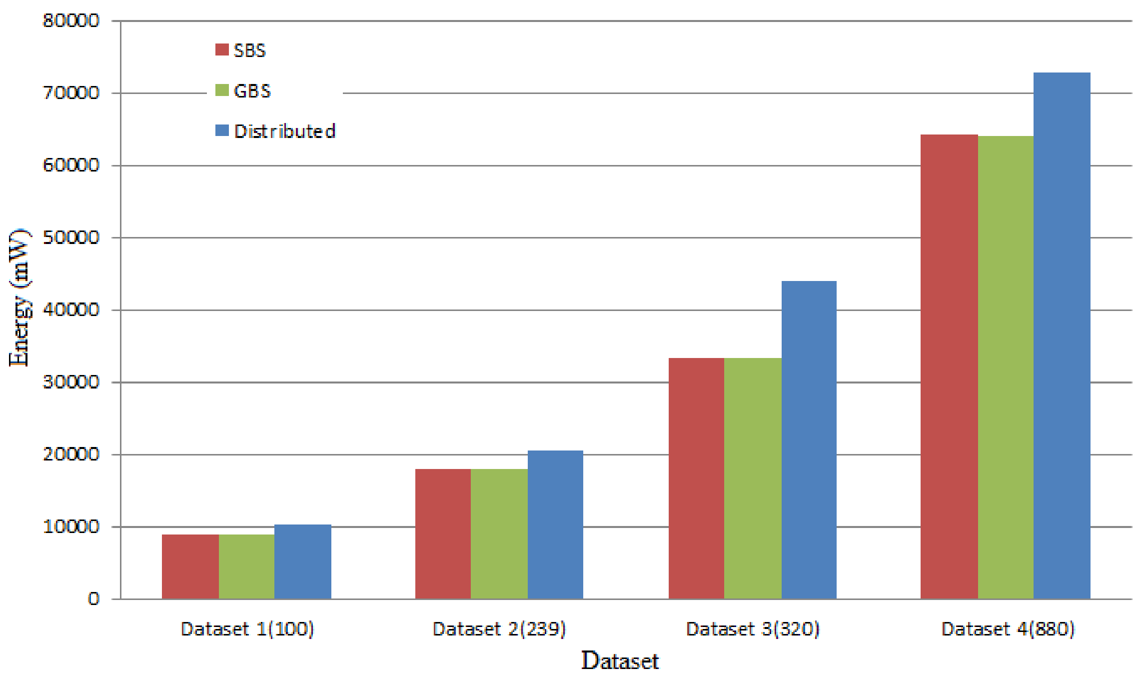

Figure 5.6: Energy consumption as a result of using Distributed, SBS and GBS Algorithms

Next we experiment to study the impact of density on the size of broadcaster set. Figure 5.7 show that the size of broadcaster set is inversely proportional to the density of the network. This behaviour is observed after application of all SBS, GBS and distributed algorithm. It can be also noticed that number of broadcasters as detected by distributed algorithm is more than those found by SBS and GBS algorithm which is due to the fact that SBS and GBS aims to find optimal set of broadcasters.

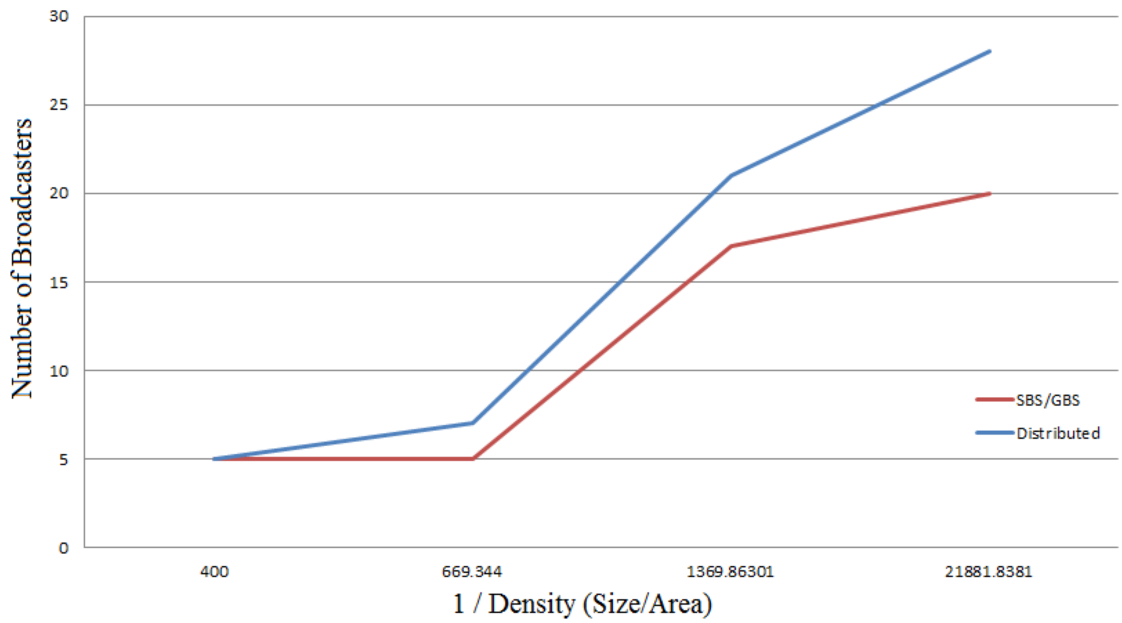

Figure 5.7: Size of broadcaster set versus Inverse of Density

In the last set of experiments, we aim to see the impact of alpha on the broadcaster probability computation. For this we take a small data set of size 50 and allocate half of the nodes (that is 25 nodes) with residual energy equal to 50% of the total smartphone energy depicted by green coloured nodes in Figure 5.8.

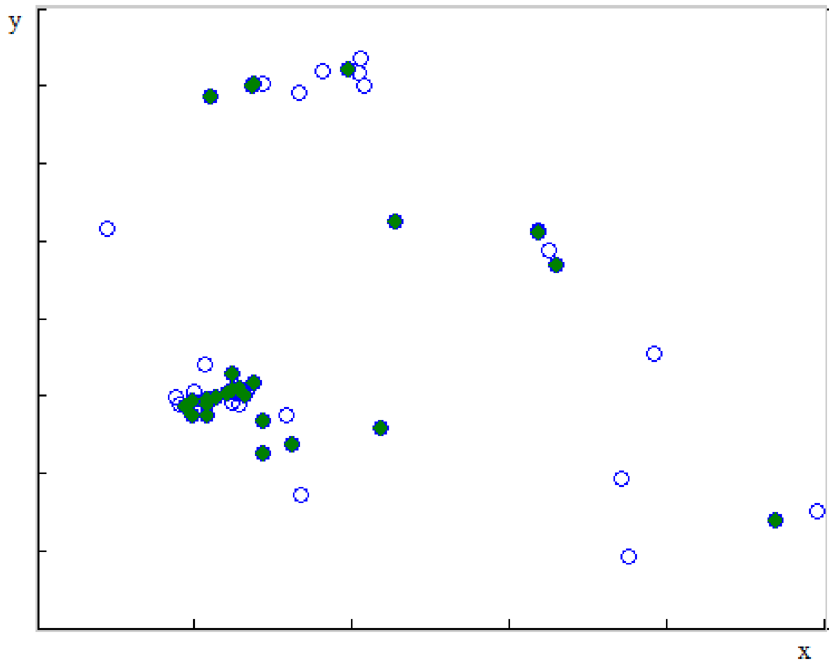

Figure 5.8: Scatter Plot of 50 Participants

We consider total energy of smartphone to be 319000 mW. Then we run distributed algorithm with $\alpha = 1$ (Figure 5.9) and $\alpha = 0$ (Figure 5.10). The red coloured nodes depict broadcasters. The squared nodes are the ones which change their role as alpha changes. It is clear from Figure 5.9, that node with high connectivity is given preference and in Figure 5.10, nodes with higher energy are given preference.

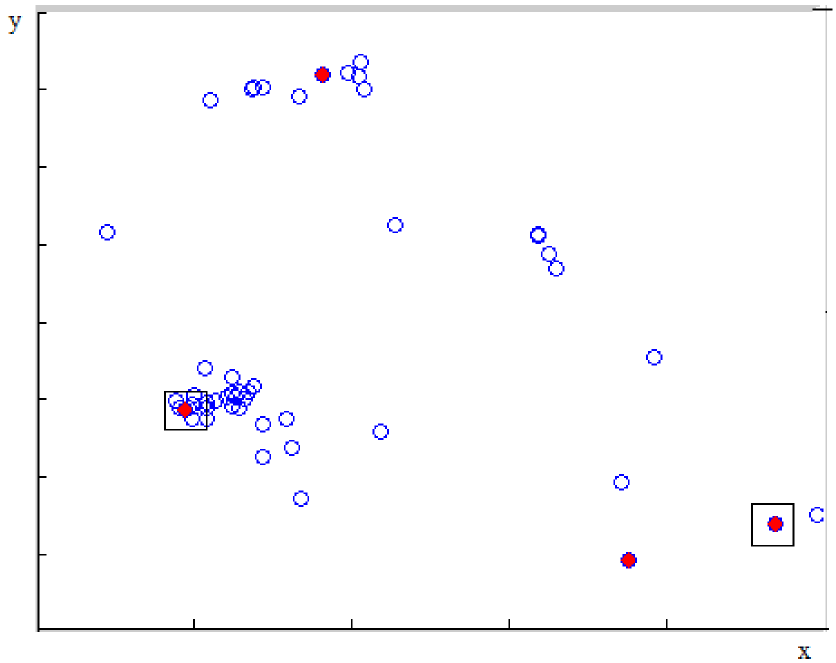

Figure 5.9: Broadcasters when $\alpha = 1$

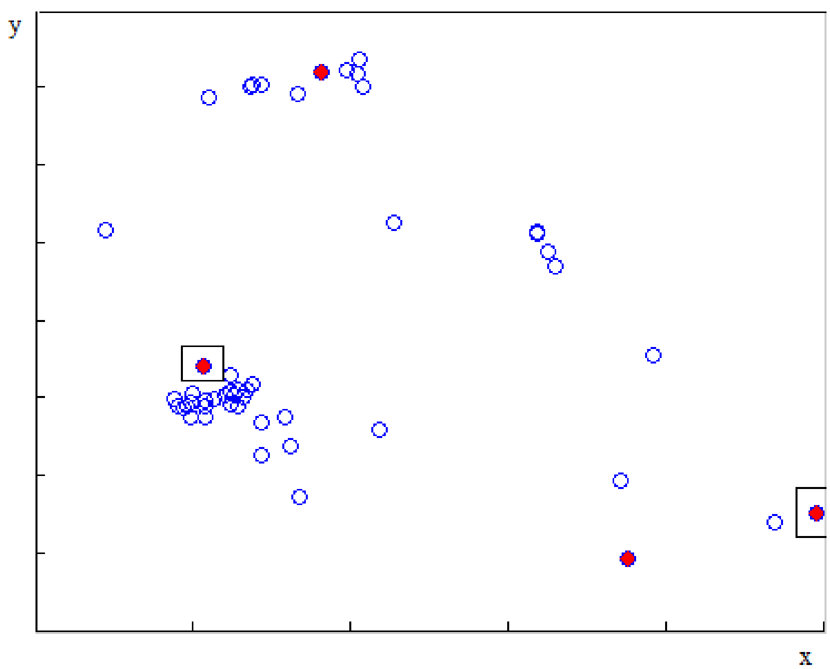

Figure 5.10: Broadcasters when $\alpha = 0$

5.7 Conclusion

We provided a distributed approach where every participant finds their role by themselves based on local information. This is first work to the best of our knowledge for assigning role to participants without involving server. We also considered the residual energy of devices for the selection of broadcaster which has not been considered in the centralized approach. The experiments validate that distributed approach takes 70-85% less time than centralized approach.

Chapter 6

Conclusion and future work

6.1 Contributions

We provide an enhancement to the existing work on participatory sensing system. We have suggested a modification to energy model which is capable of minimizing power consumption of the system and proposed a sorting based algorithm for broadcaster set selection problem. This proves to be better for data sets of moderate and higher sizes which is actually the scenario in real cases.

The original GBS algorithm is not very suitable for the dynamic scenarios especially when number of participants is quite large. It is not suitable to rerun the brute-force algorithm on entire database for each insertion or deletion of participant. In lieu of that concern, we proposed incremental algorithms that help in saving lot of time. Experimental results validate efficiency of the algorithms.

Apart from this, we have provided a distributed approach where every participant finds their role by themselves. This is first work to the best of our knowledge for assigning role to participants without involving server and devices make decision based on local information. In addition to this, we also considered residual energy of the devices for the selection of broadcasters. Results show that distributed approach takes much less time than centralized approach although it is not capable of finding optimal set of broadcasters.

6.2 Future work

Considering mobility, relative movement and direction of nodes for finding optimal set of broadcaster is part of our future work. We also plan to take into account these parameters to compute broadcaster probability and find broadcaster set in distributed approach. Periodic computation of role is not an efficient technique. So, there is a need of a protocol by which nodes can compute period of localization and reassignment of role in distributed scheme. We have set a threshold of 0.01 for broadcaster probability which was used for all experiments in Chapter 5. However, in future we focus to find an optimal value for the same and hope to minimize the energy difference in time efficient way.

Even though we have tried to minimize power needs of the devices, still there are chances that devices might refrain from participating in sensing activity. So, there is a need of incentive scheme to receive good participation from devices.

Privacy and security is one of the primary concerns in mobile sensing. We plan to include effective mechanisms to make system and participating devices secure. In addition to this, there is also a possibility when a node or group of devices get compromised. This leads to inconsistency of aggregated data and false information which may even destroy security of legitimate nodes. For this it is essential to detect attackers and this can be easily extended to the existing framework based on the information uploaded by broadcasters and normal participants.

References

[1] E. Macias, A. Suarez, J. Lloret. "Mobile Sensing Systems." *Sensors*, pp. 17292–17321, 2013.

[2] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury and A.T. Campbell. "A Survey of Mobile Phone Sensing." *IEEE Communications magazine*, vol. 48, pp. 140–150, 2010.

[3] N.D. Lane, S.B. Eisenman, M. Musolesi, E. Miluzzo, A.T. Campbell. "Urban sensing systems: opportunistic or participatory?" *Proceedings of the 9th workshop on Mobile computing systems and applications.*, pp. 11–16, 2008.

[4] Intel/UC Berkeley. "Urban Atmospheres." Available at http://www.urban-atmospheres.net/.

[5] P. Dutta, P. Aoki , N. Kumar , A. Mainwaring, C. Myers, W. Willett, A. Woodruff. "Common Sense: participatory urban sensing using a network of handheld air quality monitors." *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 349–350, 2009.

[6] E. Kanjo. "NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping." *Mobile Networks and Applications.*, vol. 15, pp. 562–574, 2010.

[7] Z. Zhang, T. Liu, D. Chen and W. Zhang. "Novel Algorithm for Identifying and Fusing Conflicting Data in Wireless Sensor Networks." *Sensors*, vol. 14, pp. 9562–9581, 2014.

[8] J.H. Ahnn, U. Lee, H.J. Moon. "Geoserv: A distributed urban sensing platform." *Cluster, Cloud and Grid Computing (CCGrid)*, pp. 164–173, IEEE 2011.

[9] H. Lua, N.D. Lanea, S.B. Eisenmanb, A.T. Campbell. "Bubble-sensing: Binding sensing tasks to the physical world." *Pervasive and Mobile Computing*, pp. 58–71, 2010.

[10] Q. Chen, Z. Weng, Y. Han, Y. Zhu. "A distributed algorithm for maximizing utility of data collection in a crowd sensing system." *International Journal of Distributed Sensor Networks*, vol. 12, no. 9, pp. 1–10, 2016.

[11] Z. Song, b. Zhang, C.H. Liu, W. Wang. "QoI-Aware Energy-Efficient Participant Selection." *Proceedings of the IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Singapore*, pp. 248–256, 2014.

[12] G.S. Tuncay, G. Benincasa, A. Helmy. Autonomous and distributed recruitment and data collection framework for opportunistic sensing. *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, pp. 50–53, 2013.

[13] S. Reddy, D. Estrin, M. Srivastava. "Recruitment Framework for Participatory Sensing Data Collections." *Pervasive Computing*, vol. 6030, pp. 138–155, 2010.

[14] H. Li, T. Li, Y. Wang. "Dynamic Participant Recruitment of Mobile Crowd Sensing for Heterogeneous Sensing Tasks." *Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 136–144, 2015.

[15] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, Andrew T. "The Jigsaw continuous sensing engine for mobile phone applications." *SenSys '10 Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 71–84, 2010.

[16] I. Shafer, M.L. Chang. "Movement detection for power-efficient smartphone WLAN localization." *Modeling, analysis, and simulation of wireless and mobile systems*, pp. 81–90, 2010.

[17] P. Baier, F. Durr, K. Rothermel. "PSense: Reducing Energy Consumption in Public Sensing Systems." *Advanced Information Networking and Applications (AINA-2012)*, pp. 136–143, 2012.

[18] C.H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, K.K. Leung. "Energy-Aware Participant Selection for Smartphone-Enabled Mobile Crowd Sensing." *IEEE Systems Journal*, pp. 1–12, 2015.

[19] W. Wang, T. Xi, E.C. Ngai, Z. Song. "Energy-Efficient Collaborative Outdoor Localization for Participatory Sensing." *Sensors*, vol. 16, pp. 1–21, 2016.

[20] T.A. Johnson, P. Seeling. "Localization using bluetooth device names." *Mobile Ad Hoc Networking and Computing (MobiHoc '12)*, pp. 247–248, 2012.

[21] B.Xiao, S.Zhou. "Distributed localization using a moving beacon in wireless sensor networks." *IEEE Transaction on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 587–600, 2008.

[22] B, Zhang, Fengqi, Z. Zhang. "Collaborative localization algorithm for wireless sensor networks using mobile anchors." *Computational Intelligence and Industrial Applications(PACIIA)*, pp. 309–312, 2009.

[23] Z. Song, J. Ma, M. Dong, W. Wang, X. Gong, X. Que. "Phone-Radar: Infrastructure-free Device-to-device Localization." *Vehicular Technology Conference (VTC Spring)*, 2014

[24] A. Bose, C.H. Foh. "A Practical Path Loss Model For Indoor WiFi Positioning Enhancement." *Information, Communications and Signal Processing*, pp. 15, 2007.

[25] T. Xi, W. Wang, E.C.H. Ngai, Z. Song, Y. Tian, X. Gong. "Energy-Efficient Collaborative Localization for Participatory Sensing System." *Global Communications Conference(GLOBECOM)*, pp. 1–6, IEEE 2015.

[26] M. Ester,H.P. Kriegel,J. Sander,M. Wimmer,X. Xu. "Incremental clustering for mining in a data ware housing." *VLDB '98 Proceedings of the 24rd International Conference on Very Large Data Bases*, pp. 323–333, 1998.

[27] I. Constandache, S. Gaonkar, M. Sayler, R.R. Choudhury, L. Cox. "Enloc: Energy-efficient localization for mobile phones." *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2716-2720, 2009.

[28] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha. "AppScope: Application Energy Metering Framework for Android Smartphone Using Kernel Activity Monitoring." *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, pp. 387-400, 2012.

[29] K.A. Delin, S.P. Jackson, and R.R. Some. "Sensor Webs." *NASA Tech Briefs*, 23, 90, 1999.

[30] R.K. Ganti, F. Ye, H. Lei. "Mobile Crowdsensing: Current State and Future Challenges." *IEEE Communications Magazine*, vol. 49, pp. 32–39, 2011.

[31] K.A. Delin. "The Sensor Web: A Macro-Instrument for Coordinated Sensing." *Sensors*, pp. 270–285, 2002.

[32] B. Guo, Z. Yu, X. Zhou, D. Zhang. "From Participatory Sensing to Mobile Crowd Sensing. " *PerCom Workshops*, pp. 593–598, 2014.

[33] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M.B. Srivastava. "Participatory Sensing." *World Sensor Web Workshop, ACM Sensys*, 2006.

[34] D. Estrin. "Participatory Sensing: Applications and Architecture." *MobiSys*, pp. 3–4, 2010.

[35] J. Goldman, K. Shilton, J. Burke, et al. "Participatory sensing a citizen-powered approach to illuminating the patterns that shape our world." $http://www.wilsoncenter.org/sites/default/files/participatory_sensing.pdf$.

[36] A. Kailas, C.C. Chong, F. Watanabe. "From Mobile Phones to Personal Wellness Dashboards." *IEEE Pulse*, vol. 1, no. 1, pp. 57-63, 2010.

[37] Y. Zheng, X. Xi, W.Y. Ma. "GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory." *IEEE Data Eng. Bull.*, vol. 33, pp. 32–39, 2010.

[38] J. Hicks, N. Ramanathan, D. Kim, M. Monibi, J. Selsky, M. Hansen, D. Estrin. "AndWellness: An Open Mobile System for Activity and Experience Sampling." *Proceedings of the 1st Wireless Health Conference*, pp. 34-43, 2010.

[39] J. Ryder, B. Longstaff, S. Reddy, D. Estrin. "Ambulation: a tool for monitoring mobility patterns over time using mobile phones." *Proceedings of the IEEE International Conference on Social Computing*, vol. 4, pp. 927-931, 2009.

[40] S.B. Eisenman, E. Miluzzo, N.D. Lane, R.A. Peterson, G-S. Ahn, A.T. Campbell. "The BikeNet Mobile Sensing System for Cyclist Experience Mapping" *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems*, pp. 87-101, 2007.

[41] H. Lu; M. Rabbi; G.T. Chittaranjan; et al. "Stresssense: detecting stress in unconstrained acoustic environments using smartphones." *Proceedings of the ACM UbiComp*, pp. 351–360, 2012.

[42] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin and M. Srivastava. "Biketastic: sensing and mapping for better biking." *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems*, pp. 1817-1820, 2010.

[43] M. Kaenel, P. Sommer and R. Wattenhofer. "Ikarus: Large-Scale Participatory Sensing at High Altitudes." *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, pp. 63-68, 2011.

[44] E. Miluzzo, N.D. Lane, K. Fodor et al. "Sensing meets mobile social networks: the design; implementation and evaluation of the cenceme application." *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 337-350, 2008.

[45] L. Deng and L.P. Cox. "Live compare: grocery bargain hunting through participatory sensing." *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications*, pp. 1-6, 2009

[46] D. Peebles, H. Lu, N.D. Lane, T. Choudhury and A.T. Campbell. "Community-guided learning: exploiting mobile sensor users to model human behavior." *Proceedings of the 24th AAAI Conference on Artificial Intelligence and the 22nd Innovative Applications of Artificial Intelligence Conference*, pp. 1600-1606, 2010.

[47] S. Mathur, T. Jin, N. Kasturirangan et al. "ParkNet: drive-by sensing of road-side parking statistics." *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services*, pp. 123-136, 2010.

[48] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden and H. Balakrishnan. "The Pothole Patrol: using a mobile sensor network for road surface monitoring." *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, pp. 29-39, 2008.